

Synthesis of Memory-Efficient Real-Time Controllers for Safety Objectives ^{*}

Krishnendu Chatterjee¹ and Vinayak S. Prabhu²

¹ Institute of Science and Technology (IST) Austria

² University of Porto

krish.chat@ist.ac.at, vinayak@eecs.berkeley.edu

Abstract. We study synthesis of controllers for real-time systems, where the objective is to stay in a given safe set. The problem is solved by obtaining winning strategies in concurrent two-player *timed automaton games* with safety objectives. To prevent a player from winning by blocking time, we restrict each player to strategies that ensure that the player cannot be responsible for causing a zeno run. We construct winning strategies for the controller which require access only to (1) the system clocks (thus, controllers which require their own internal infinitely precise clocks are not necessary), and (2) a linear (in the number of clocks) number of memory bits. Precisely, we show that a memory of size $(3 \cdot |C| + 1 + \lg(|C| + 1))$ bits suffices for winning controller strategies for safety objectives, where C is the set of clocks of the timed automaton game, significantly improving the previous known exponential bound. We also settle the open question of whether *region* strategies for controllers require memory for safety objectives by showing with an example that region strategies do require memory for safety objectives.

1 Introduction

Synthesizing controllers to ensure that a plant stays in a safe set is an important problem in the area of systems control. We study the synthesis of *timed* controllers in the present paper. Our formalism is based on timed automata [AD94], which are models of real-time systems in which states consist of discrete locations and values for real-time clocks. The transitions between locations are dependent on the clock values. The real-time controller synthesis problem is modeled using *timed automaton games*, which are played by two players on timed automata, where player 1 is the “controller” and player 2 the “plant”. Obtaining winning strategies for player 1 in such games corresponds to the construction of controllers for real-time systems with desired objectives.

The issue of *time divergence* is crucial in timed games, as a naive control strategy might simply block time, leading to “zeno” runs. The following approaches have been proposed to avoid such invalid zeno solutions: (1) discretize time so that players can only take transitions at integer multiples of some fixed time period, e.g. in [HK99]; (2) put syntactic restrictions on the timed game structure so that zeno runs are not possible (the syntactic restriction is usually presented as the *strong non-zenoness* assumption where the obtained controller synthesis algorithms are guaranteed to work correctly only on timed automaton games where every cycle is such that in it some clock is reset to 0 and is also greater than an integer value at some point, e.g. in [AM99,BBL04,PAMS98]); (3) require player 1 to ensure time divergence (e.g. by only taking transitions if player 2 can never take transitions in the future from the current location, as in [DM02,BDMP03]); (4) give the controller access to an extra (infinitely precise) clock which measure global time and require that player 1 wins if either its moves are chosen only finitely often, or if the ticks of this extra clock are seen infinitely often while satisfying the desired objective, e.g. in [dAFH⁺03,AH97].

The above approaches are not optimal in many cases and below we point out some drawbacks. Discretizing the system blows up the state space; and might not be faithful to the real-time semantics. Putting syntactic restrictions is troublesome as it can lead to disallowing certain system models. For example, consider the timed automaton game \mathcal{T} in Figure 1. The details of the game are not important and are omitted here for the sake of brevity. In the figure, the edges are labelled as a_1^j for actions controlled by player 1; and by a_2^j for actions controlled by player 2. The safety objective is to avoid the location “Bad” (player 1 can satisfy this objective

^{*} This work has been financially supported in part by the European Community’s Seventh Framework Programme via project Control for coordination of distributed systems (C4C; Grant Agreement number INFSo-ICT-223844); and by Austrian FWF NFN ARiSE funding.

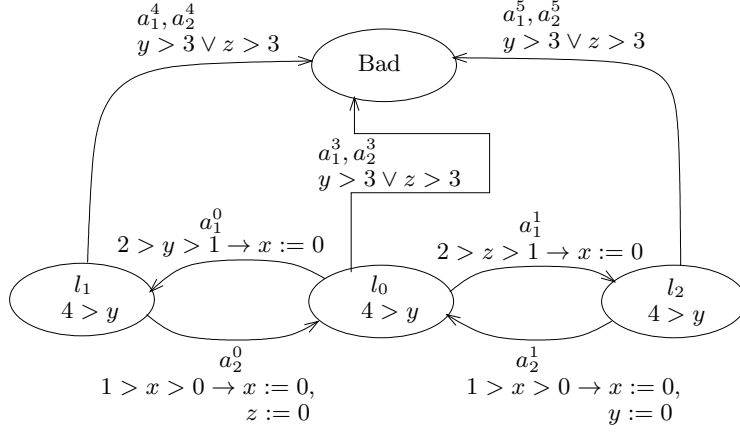


Fig. 1. A timed automaton game.

without blocking time). One can easily show that zeno runs are possible in this timed automaton game, mainly, due to the edges a_2^0 and a_1^1 . The game \mathcal{T} can be made to be non-zeno syntactically by changing the guards of the edges a_2^0 and a_1^1 to $1 > x > d$, where d is some conservative constant (say 0.001 time units, where it is assumed that the plant takes at least 0.001 time units to transition out of l_1 and l_2). This change unfortunately blows up the finite state *region abstraction* of the timed automaton game (the region abstraction is used in every current solution to the real-time controller synthesis problem for timed automaton games). If the constant d is 0.001, then the number of states in the region abstraction blows up from roughly $2.5 * 10^5$ for the original game to $2.5 * 10^5 * 10^9$; a blow up by a *factor* of 10^9 . Admittedly however, on the fly algorithms for controller synthesis may help mitigate the situation in some cases ([CDF⁺05]) by not explicitly constructing the full graph of the region abstraction.

Requiring player 1 to guarantee time divergence by only taking transitions if player 2 cannot take transitions from the current location is too conservative. If we consider the game in Figure 1, this approach would prevent player 1 from taking *any* of the actions, making the system uncontrollable. Finally, adding an extra infinitely precise clock to measure time, and making it observable to the controller amounts to giving unfair and unrealistic power to the controller in many situations.

In the present paper, we avoid the shortcomings of the previous approaches by using two techniques. First, we use *receptive* [AH97,SGSAL98], player-1 strategies, which, while being required to not prevent time from diverging, are not required to ensure time divergence. Receptiveness is incorporated by using the more general, semantic and fully symmetric formalism of [dAFH⁺03] for dealing with the issue of time divergence. This setting places no syntactic restriction on the game structure, and gives both players equally powerful options for advancing time, but for a player to win, it must not be *responsible* for causing time to converge. Formally, our timed games proceed in an infinite sequence of rounds. In each round, both players simultaneously propose moves, with each move consisting of an action and a time delay after which the player wants the proposed action to take place. Of the two proposed moves, the move with the shorter time delay “wins” the round and determines the next state of the game. Let a set Φ of runs be the desired objective for player 1. Then player 1 has a *winning* strategy for Φ if it has a strategy to ensure that, no matter what player 2 does, one of the following two conditions hold: (1) time diverges and the resulting run belongs to Φ , or (2) time does not diverge but player-1’s moves are chosen only finitely often (and thus it is not to be blamed for the convergence of time). Second, in the current work, the controller only uses the system clocks of the model (unlike [dAFH⁺03] which makes available to the controller an extra infinitely precise clock to measure time), ensuring that the controller bases its actions only on the variables corresponding to the physical processes of the system (the system clocks). Time divergence is *inferred* from the history of certain predicates of the system clocks, rather than from an extra infinitely precise clock that the controller has to keep in memory.

Contributions. Our current work significantly improves the results of [CHP08]. In [CHP08] we showed that finite-memory receptive strategies suffice for safety objective in timed automaton games; the problem of establishing a memory bound was left open. In this paper, we first show that a basic analysis using *Zielonka trees* of the characterization of receptive strategies of [CHP08] leads to an *exponential* number of bits for the memory bound (in the number of clocks) for the winning strategies. We then present an improved new characterization of receptive strategies for safety objectives which allows us to obtain a *linear* number of bits for the memory bound for winning strategies. Precisely, we show that a memory of size $(3 \cdot |C| + 1 + \lg(|C| + 1))$ bits suffices for winning receptive strategies for safety objectives, where C is the set of clocks of the timed automaton game, considerably improving the exponential bound obtained from the previous result. Finally, we settle the open question of whether *region* strategies for controllers require memory for safety objectives. We show with an example that region strategies in general do require memory for safety objectives.

2 Timed Games

2.1 Timed Game Structures

In this Subsection we present the definitions of timed game structures, runs, objectives, strategies and the notions of sure and almost-sure winning in timed game structures.

Timed game structures. A *timed game structure* is a tuple $\mathcal{G} = \langle S, A_1, A_2, \Gamma_1, \Gamma_2, \delta \rangle$ with the following components.

- S is a set of states.
- A_1 and A_2 are two disjoint sets of actions for players 1 and 2, respectively. We assume that $\perp_i \notin A_i$, and write A_i^\perp for $A_i \cup \{\perp_i\}$. The set of *moves* for player i is $M_i = \mathbb{R}_{\geq 0} \times A_i^\perp$. Intuitively, a move $\langle \Delta, a_i \rangle$ by player i indicates a waiting period of Δ time units followed by a discrete transition labeled with action a_i . The move $\langle \Delta, \perp_i \rangle$ is used to represent the move of player i where player- i just lets time elapse for Δ time units without taking any of the discrete actions from A_i .
- $\Gamma_i : S \mapsto 2^{M_i} \setminus \emptyset$ are two move assignments. At every state s , the set $\Gamma_i(s)$ contains the moves that are available to player i . We require that $\langle 0, \perp_i \rangle \in \Gamma_i(s)$ for all states $s \in S$ and $i \in \{1, 2\}$. Intuitively, $\langle 0, \perp_i \rangle$ is a time-blocking stutter move.
- $\delta : S \times (M_1 \cup M_2) \mapsto S$ is the transition function. We require that for all time delays $\Delta, \Delta' \in \mathbb{R}_{\geq 0}$ with $\Delta' \leq \Delta$, and all actions $a_i \in A_i^\perp$, we have (1) $\langle \Delta, a_i \rangle \in \Gamma_i(s)$ iff both $\langle \Delta', \perp_i \rangle \in \Gamma_i(s)$ and $\langle \Delta - \Delta', a_i \rangle \in \Gamma_i(\delta(s, \langle \Delta', \perp_i \rangle))$; and (2) if $\delta(s, \langle \Delta', \perp_i \rangle) = s'$ and $\delta(s', \langle \Delta - \Delta', a_i \rangle) = s''$, then $\delta(s, \langle \Delta, a_i \rangle) = s''$.

The game proceeds as follows. If the current state of the game is s , then both players simultaneously propose moves $\langle \Delta_1, a_1 \rangle \in \Gamma_1(s)$ and $\langle \Delta_2, a_2 \rangle \in \Gamma_2(s)$. The move with the shorter duration “wins” in determining the next state of the game. If both moves have the same duration, then player 2 determines whether the next state will be determined by its move, or by the move of player 1. We use this setting as our goal is to compute the winning set for player 1 against all possible strategies of player 2. Formally, we define the *joint destination function* $\delta_{jd} : S \times M_1 \times M_2 \mapsto 2^S$ by

$$\delta_{jd}(s, \langle \Delta_1, a_1 \rangle, \langle \Delta_2, a_2 \rangle) = \begin{cases} \{\delta(s, \langle \Delta_1, a_1 \rangle)\} & \text{if } \Delta_1 < \Delta_2; \\ \{\delta(s, \langle \Delta_2, a_2 \rangle)\} & \text{if } \Delta_2 < \Delta_1; \\ \{\delta(s, \langle \Delta_2, a_2 \rangle), \delta(s, \langle \Delta_1, a_1 \rangle)\} & \text{if } \Delta_2 = \Delta_1. \end{cases}$$

The time elapsed when the moves $m_1 = \langle \Delta_1, a_1 \rangle$ and $m_2 = \langle \Delta_2, a_2 \rangle$ are proposed is given by $\text{delay}(m_1, m_2) = \min(\Delta_1, \Delta_2)$. The boolean predicate $\text{blame}_i(s, m_1, m_2, s')$ indicates whether player i is “responsible” for the state change from s to s' when the moves m_1 and m_2 are proposed. Denoting the opponent of player i by $\sim i = 3 - i$, for $i \in \{1, 2\}$, we define

$$\text{blame}_i(s, \langle \Delta_1, a_1 \rangle, \langle \Delta_2, a_2 \rangle, s') = (\Delta_i \leq \Delta_{\sim i} \wedge \delta(s, \langle \Delta_i, a_i \rangle) = s').$$

Runs. A *run* of the timed game structure \mathcal{G} is an infinite sequence $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \dots$ such that $s_k \in S$ and $m_i^k \in \Gamma_i(s_k)$ and $s_{k+1} \in \delta_{\text{jd}}(s_k, m_1^k, m_2^k)$ for all $k \geq 0$ and $i \in \{1, 2\}$. For $k \geq 0$, let $\text{time}(r, k)$ denote the “time” at position k of the run, namely, $\text{time}(r, k) = \sum_{j=0}^{k-1} \text{delay}(m_1^j, m_2^j)$ (we let $\text{time}(r, 0) = 0$). By $r[k]$ we denote the $(k+1)$ -th state s_k of r . The run prefix $r[0..k]$ is the finite prefix of the run r that ends in the state s_k . Let Runs be the set of all runs of \mathcal{G} , and let FinRuns be the set of run prefixes.

Objectives. An *objective* for the timed game structure \mathcal{G} is a set $\Phi \subseteq \text{Runs}$ of runs. We will be interested in the classical safety objectives. Given a set of states Y , the *safety* objective consists of the set of runs that stay within Y , formally, $\text{Safe}(Y) = \{r \mid \text{for all } i \text{ we have } r[i] \in Y\}$. To solve timed games for safety objectives, we shall need to solve for certain ω -regular objectives (see [Tho97] for the definition of ω -regular sets).

Strategies. A *strategy* for a player is a recipe that specifies how to extend a run. Formally, a *probabilistic strategy* π_i for player $i \in \{1, 2\}$ is a function π_i that assigns to every run prefix $r[0..k]$ a probability measure $P_{\pi_i}^{r[0..k]}$ over $\Gamma_i(r[k])$, the set of moves available to player i at the state $r[k]$ (the event class can be suitably chosen). *Pure strategies* are strategies for which the state space of the probability distribution of $P_{\pi_i}^{r[0..k]}$ is a singleton set for every run r and all k . We let Π_i^{pure} denote the set of pure strategies for player i , with $i \in \{1, 2\}$. We call probability distributions with singleton support sets as *pure distributions*.

For $i \in \{1, 2\}$, let Π_i be the set of strategies for player i . If both both players propose the same time delay, then the tie is broken by a *scheduler*. Let TieBreak be the set of functions from $\mathbb{R}_{\geq 0} \times A_1^\perp \times A_2^\perp$ to $\{1, 2\}$. A *scheduler strategy* π_{sched} is a mapping from FinRuns to TieBreak . If $\pi_{\text{sched}}(r[0..k]) = h$, then the resulting state given player 1 and player 2 moves $\langle \Delta, a_1 \rangle$ and $\langle \Delta, a_2 \rangle$ respectively, is determined by the move of player $h(\Delta, a_1, a_2)$. We denote the set of all scheduler strategies by Π_{sched} . Given two strategies $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$, the set of possible *outcomes* of the game starting from a state $s \in S$ is denoted $\text{Outcomes}(s, \pi_1, \pi_2)$. We let $\text{Outcomes}_k(s, \pi_1, \pi_2)$ denote the set of finite runs $r[0..k-1]$ which are possible according to the two strategies given the initial state s . If we fix the scheduler strategy π_{sched} then the set of possible outcomes is denoted by $\text{Outcomes}(s, \pi_1, \pi_2, \pi_{\text{sched}})$. Given strategies π_1 and π_2 , for player 1 and player 2, respectively, a scheduler strategy π_{sched} and a starting state s we denote by $\text{Pr}_s^{\pi_1, \pi_2, \pi_{\text{sched}}}(\cdot)$ the probability space over Runs given the strategies and the initial state s .

Receptive strategies. We will be interested in strategies that are meaningful (in the sense that they do not block time). To define them formally we first present the following two sets of runs.

- A run r is *time-divergent* if $\lim_{k \rightarrow \infty} \text{time}(r, k) = \infty$. We denote by Timediv the set of all time-divergent runs.
- The set $\text{Blameless}_i \subseteq \text{Runs}$ consists of the set of runs in which player i is responsible only for finitely many transitions. A run $s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \dots$ belongs to the set Blameless_i , for $i = \{1, 2\}$, if there exists a $k \geq 0$ such that for all $j \geq k$, we have $\neg \text{blame}_i(s_j, m_1^j, m_2^j, s_{j+1})$.

A strategy π_i is *receptive* if for all strategies $\pi_{\sim i}$, all states $s \in S$, and all runs $r \in \text{Outcomes}(s, \pi_1, \pi_2)$, either $r \in \text{Timediv}$ or $r \in \text{Blameless}_i$. Thus, no what matter what the opponent does, a receptive strategy of player i cannot be responsible for blocking time. Strategies that are not receptive are not physically meaningful. A timed game structure \mathcal{G} is *well-formed* if both players have receptive strategies. We restrict our attention to well-formed timed game structures. We denote Π_i^R to be the set of receptive strategies for player i . Note that for $\pi_1 \in \Pi_1^R, \pi_2 \in \Pi_2^R$, we have $\text{Outcomes}(s, \pi_1, \pi_2) \subseteq \text{Timediv}$.

Sure and almost-sure winning modes. Let $\text{Sure}_1^{\mathcal{G}}(\Phi)$ (resp. $\text{AlmostSure}_1^{\mathcal{G}}(\Phi)$) be the set of states s in \mathcal{G} such that player 1 has a receptive strategy $\pi_1 \in \Pi_1^R$ such that for all scheduler strategies $\pi_{\text{sched}} \in \Pi_{\text{sched}}$ and for all player-2 receptive strategies $\pi_2 \in \Pi_2^R$, we have $\text{Outcomes}(s, \pi_1, \pi_2) \subseteq \Phi$ (resp. $\text{Pr}_s^{\pi_1, \pi_2, \pi_{\text{sched}}}(\Phi) = 1$). Such a winning strategy is said to be a *sure* (resp. *almost sure*) winning receptive strategy. In computing the winning sets, we shall quantify over *all* strategies, but modify the objective to take care of time divergence. Given an objective Φ , let $\text{TimeDivBl}_1(\Phi) = (\text{Timediv} \cap \Phi) \cup (\text{Blameless}_1 \setminus \text{Timediv})$, i.e., $\text{TimeDivBl}_1(\Phi)$ denotes the set of paths such that either time diverges and Φ holds, or else time converges and player 1 is not responsible for time to converge. A player-1 strategy is hence receptive iff it ensures that against all player-2 strategies, the resulting runs belong to $\text{TimeDivBl}_1(\text{Runs})$. Let $\underline{\text{Sure}}_1^{\mathcal{G}}(\Phi)$ (resp. $\underline{\text{AlmostSure}}_1^{\mathcal{G}}(\Phi)$) be the set of states in \mathcal{G} such that for all $s \in \underline{\text{Sure}}_1^{\mathcal{G}}(\Phi)$ (resp. $\underline{\text{AlmostSure}}_1^{\mathcal{G}}(\Phi)$), player 1 has a strategy $\pi_1 \in \Pi_1$ such that for all strategies for

all scheduler strategies $\pi_{\text{sched}} \in \Pi_{\text{sched}}$ and for all player-2 strategies $\pi_2 \in \Pi_2$, we have $\text{Outcomes}(s, \pi_1, \pi_2) \subseteq \Phi$ (resp. $\Pr_s^{\pi_1, \pi_2, \pi_{\text{sched}}}(\Phi) = 1$). Such a winning strategy is said to be a sure (resp. almost sure) winning for the non-receptive game. The following result establishes the connection between **Sure** and **Sure** sets.

Theorem 1 ([HP06]). *For all well-formed timed game structures \mathcal{G} , and for all ω -regular objectives Φ , we have $\underline{\text{Sure}}_1^{\mathcal{G}}(\text{TimeDivBl}_1(\Phi)) = \text{Sure}_1^{\mathcal{G}}(\Phi)$.*

We observe here that $\text{TimeDivBl}_1(\Phi)$ is *not* equivalent to $(\neg \text{Blameless}_1) \rightarrow \text{TimeDiv} \cap \Phi$. Player 1 loses even if it does not get moves infinitely often, provided time diverges and the run does not belong to Φ .

2.2 Timed Automaton Games

In this Subsection we define a special class of timed game structures, namely, timed automaton games, and the notion of region equivalence.

Timed automaton games. Timed automata [AD94] suggest a finite syntax for specifying infinite-state timed game structures. A *timed automaton game* is a tuple $\mathcal{T} = \langle L, C, A_1, A_2, E, \gamma \rangle$ with the following components:

- L is a finite set of locations.
- C is a finite set of clocks.
- A_1 and A_2 are two disjoint sets of actions for players 1 and 2, respectively.
- $E \subseteq L \times (A_1 \cup A_2) \times \text{Constr}(C) \times L \times 2^C$ is the edge relation, where the set $\text{Constr}(C)$ of *clock constraints* is generated by the grammar

$$\theta ::= x \leq d \mid d \leq x \mid \neg \theta \mid \theta_1 \wedge \theta_2$$

for clock variables $x \in C$ and nonnegative integer constants d . For an edge $e = \langle l, a_i, \theta, l', \lambda \rangle$, the clock constraint θ acts as a guard on the clock values which specifies when the edge e can be taken, and by taking the edge e , the clocks in the set $\lambda \subseteq C$ are reset to 0. We require that for all edges $\langle l, a_i, \theta', l', \lambda' \rangle, \langle l, a_i, \theta'', l'', \lambda'' \rangle \in E$ with $l' \neq l''$, the conjunction $\theta' \wedge \theta''$ is unsatisfiable. This requirement ensures that a state and a move together uniquely determine a successor state.

- $\gamma : L \mapsto \text{Constr}(C)$ is a function that assigns to every location an invariant for both players. All clocks increase uniformly at the same rate. When at location l , each player i must propose a move out of l before the invariant $\gamma(l)$ expires. Thus, the game can stay at a location only as long as the invariant is satisfied by the clock values.

A *clock valuation* is a function $\kappa : C \mapsto \mathbb{R}_{\geq 0}$ that maps every clock to a nonnegative real. The set of all clock valuations for C is denoted by $K(C)$. Given a clock valuation $\kappa \in K(C)$ and a time delay $\Delta \in \mathbb{R}_{\geq 0}$, we write $\kappa + \Delta$ for the clock valuation in $K(C)$ defined by $(\kappa + \Delta)(x) = \kappa(x) + \Delta$ for all clocks $x \in C$. For a subset $\lambda \subseteq C$ of the clocks, we write $\kappa[\lambda := 0]$ for the clock valuation in $K(C)$ defined by $(\kappa[\lambda := 0])(x) = 0$ if $x \in \lambda$, and $(\kappa[\lambda := 0])(x) = \kappa(x)$ if $x \notin \lambda$. A clock valuation $\kappa \in K(C)$ *satisfies* the clock constraint $\theta \in \text{Constr}(C)$, written $\kappa \models \theta$, if the condition θ holds when all clocks in C take on the values specified by κ . A *state* $s = \langle l, \kappa \rangle$ of the timed automaton game \mathcal{T} is a location $l \in L$ together with a clock valuation $\kappa \in K(C)$ such that the invariant at the location is satisfied, that is, $\kappa \models \gamma(l)$. Let S be the set of all states of \mathcal{T} . In a state, each player i proposes a time delay allowed by the invariant map γ , together either with the action \perp , or with an action $a_i \in A_i$ such that an edge labeled a_i is enabled after the proposed time delay. We require that for $i \in \{1, 2\}$ and for all states $s = \langle l, \kappa \rangle$, if $\kappa \models \gamma(l)$, either $\kappa + \Delta \models \gamma(l)$ for all $\Delta \in \mathbb{R}_{\geq 0}$, or there exist a time delay $\Delta \in \mathbb{R}_{\geq 0}$ and an edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ such that (1) $a_i \in A_i$ and (2) $\kappa + \Delta \models \theta$ and for all $0 \leq \Delta' \leq \Delta$, we have $\kappa + \Delta' \models \gamma(l)$, and (3) $(\kappa + \Delta)[\lambda := 0] \models \gamma(l')$. This requirement is necessary (but not sufficient) for well-formedness of the game.

The timed automaton game \mathcal{T} defines the following timed game structure $\llbracket \mathcal{T} \rrbracket = \langle S, A_1, A_2, \Gamma_1, \Gamma_2, \delta \rangle$:

- $S = \{ \langle l, \kappa \rangle \mid l \in L \text{ and } \kappa(l) \text{ satisfies } \gamma(l) \}$.
- For $i \in \{1, 2\}$, the set $\Gamma_i(\langle l, \kappa \rangle)$ contains the following elements:
 1. $\langle \Delta, \perp_i \rangle$ if for all $0 \leq \Delta' \leq \Delta$, we have $\kappa + \Delta' \models \gamma(l)$.

- 2. $\langle \Delta, a_i \rangle$ if for all $0 \leq \Delta' \leq \Delta$, we have $\kappa + \Delta' \models \gamma(l)$, $a_i \in A_i$, and there exists an edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ such that $\kappa + \Delta \models \theta$.
- The transition function δ is specified by:
 1. $\delta(\langle l, \kappa \rangle, \langle \Delta, \perp_i \rangle) = \langle l, \kappa + \Delta \rangle$.
 2. $\delta(\langle l, \kappa \rangle, \langle \Delta, a_i \rangle) = \langle l', (\kappa + \Delta)[\lambda := 0] \rangle$ for the unique edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ with $\kappa + \Delta \models \theta$.

The timed game structure $\llbracket \mathcal{T} \rrbracket$ is not necessarily well-formed, because it may contain cycles along which time cannot diverge. Well-formedness of timed automaton games can be checked in EXPTIME [HP06]. We restrict our focus to well-formed timed automaton games in this paper. We shall also restrict our attention to randomization over time — a random move of a player in a timed automaton game will consist of a distribution over time over some interval I , denoted \mathcal{D}^I , together with a discrete action a_i .

Clock region equivalence. Timed automaton games can be solved using a region construction from the theory of timed automata [AD94]. For a real $t \geq 0$, let $\text{frac}(t) = t - \lfloor t \rfloor$ denote the fractional part of t . Given a timed automaton game \mathcal{T} , for each clock $x \in C$, let c_x denote the largest integer constant that appears in any clock constraint involving x in \mathcal{T} (let $c_x = 1$ if there is no clock constraint involving x). Two clock valuations κ_1, κ_2 are said to be *region equivalent*, denoted by $\kappa_1 \cong \kappa_2$ when all the following conditions hold.

1. For all clocks x with $\kappa_1(x) \leq c_x$ and $\kappa_2(x) \leq c_x$, we have $\lfloor \kappa_1(x) \rfloor = \lfloor \kappa_2(x) \rfloor$.
2. For all clocks x, y with $\kappa_i(x) \leq c_x$ and $\kappa_i(y) \leq c_y$, we have $\text{frac}(\kappa_1(x)) \leq \text{frac}(\kappa_1(y))$ iff $\text{frac}(\kappa_2(x)) \leq \text{frac}(\kappa_2(y))$.
3. For all clocks x with $\kappa_1(x) \leq c_x$ and $\kappa_2(x) \leq c_x$, we have $\text{frac}(\kappa_1(x)) = 0$ iff $\text{frac}(\kappa_2(x)) = 0$.
4. For any clock x , $\kappa_1(x) > c_x$ iff $\kappa_2(x) > c_x$. Two states $\langle \kappa_1, l_1 \rangle$ and $\langle \kappa_2, l_1 \rangle$ are region equivalent iff $l_1 = l_2$ and $\kappa_1 \cong \kappa_2$.

A *region* R of a timed automaton game \mathcal{T} is an equivalence class of states with respect to the region equivalence relation.

Representing regions. We find it useful to sometimes denote a region R by a tuple $\langle l, h, \mathcal{P}(C) \rangle$ where

- l is a location of \mathcal{T} .
- h is a function which specifies the integer values of clocks $h : C \rightarrow (\mathbb{N} \cap [0, M])$ (M is the largest constant in \mathcal{T}).
- $\mathcal{P}(C)$ is a disjoint partition of the clocks into the tuple $\langle C_{-1}, C_0, \dots, C_n \rangle$ such that $\{C_{-1}, C_0, \dots, C_n \mid \bigoplus C_i = C, C_i \neq \emptyset \text{ for } i > 0\}$.

A state s with clock valuation κ is then in the region R when all the following conditions hold.

1. The location of s corresponds to the location of R .
2. For all clocks x with $\kappa(x) \leq c_x$, $\lfloor \kappa(x) \rfloor = h(x)$.
3. For $\kappa(x) > c_x$, $h(x) = c_x$.
4. For all pair of clocks (x, y) , with $\kappa(x) \leq c_x$ and $\kappa(y) \leq c_y$, we have $\text{frac}(\kappa(x)) < \text{frac}(\kappa(y))$ iff $x \in C_i$ and $y \in C_j$ with $0 \leq i < j$ (so, $x, y \in C_k$ with $k \geq 0$ implies $\text{frac}(\kappa(x)) = \text{frac}(\kappa(y))$).
5. For $\kappa(x) \leq c_x$, $\text{frac}(\kappa(x)) = 0$ iff $x \in C_0$.
6. $x \in C_{-1}$ iff $\kappa(x) > c_x$.

There are finitely many clock regions; more precisely, the number of clock regions is bounded by $|L| \cdot \prod_{x \in C} (c_x + 1) \cdot |C|! \cdot 2^{2|C|}$.

Region equivalent runs. For a state $s \in S$, we write $\text{Reg}(s) \subseteq S$ for the clock region containing s . For a run r , we let the *region flow sequence* $\text{Reg}(r)$ be the sequence of regions R_0, R_1, \dots which intuitively denotes the regions encountered (including those during time passage specified by moves) in r . Formally, $\text{Reg}(r)$ is the region sequence R_0, R_1, \dots is such that there exist $i_0 = 0 < i_1 < i_2 \dots$ with (1) $\text{Reg}(r[j]) = R_{i_j}$; (2) $R_{k_1} \neq R_{k_2}$ for $i_j \leq k_1 < k_2 < i_{j+1}$ for any i_j ; and (3) if $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \dots$, and $r[j+1] = \delta(r[j], m_p^j)$ (for $p \in \{0, 1\}$), with $m_p^j = \langle \Delta, a \rangle$; then $R_{i_j}, R_{i_j+1}, R_{i_j+2}, \dots, R_{i_{j+1}-1}$ are the unique regions encountered when

Δ time passes from $r[j]$. The region flow sequence of a run is unique. Two runs r, r' are *region equivalent* if (1) their region flow sequences are the same, and (2) $\text{Reg}(r[j]) = \text{Reg}(r'[j])$ for all $j \geq 0$. Region equivalence for finite runs can be defined similarly. We similarly define location equivalence for runs (note that a location flow sequence is just the sequence of locations of the states in a run). An ω -regular objective Φ is a location objective if for all location-equivalent runs r, r' , we have $r \in \Phi$ iff $r' \in \Phi$. A parity index function Ω is a location parity index function if $\Omega(s_1) = \Omega(s_2)$ whenever s_1 and s_2 have the same location. Henceforth, we shall restrict our attention to location objectives.

Region equivalent strategies. Given a strategy π , a run prefix $r[0..k]$, a region R , and an action $a_i \in A_i^\perp$, let $\mathcal{W}(\pi, r[0..k], R, a_i)$ denote the set $\{\langle \Delta, a_i \rangle \mid \langle \Delta, a_i \rangle \in \text{Support}(\pi(r[0..k])) \text{ and } \text{Reg}(r[k] + \Delta) = R\}$. A strategy π_1 is a *region strategy*, if for all run prefixes $r_1[0..k]$ and $r_2[0..k]$ such that $\text{Reg}(r_1[0..k]) = \text{Reg}(r_2[0..k])$, and for all regions R and player-1 actions $a_1 \in A_1^\perp$, we have (1) $\mathcal{W}(\pi_1, r_1[0..k], R, a_1) = \emptyset$ iff $\mathcal{W}(\pi_1, r_2[0..k], R, a_1) = \emptyset$; and (2) $P_{\pi_1}^{r_1[0..k]}(\mathcal{W}(\pi_1, r_1[0..k], R, a_1)) = P_{\pi_1}^{r_2[0..k]}(\mathcal{W}(\pi_1, r_2[0..k], R, a_1))$. The definition for player 2 strategies is analogous. Two region strategies π_1 and π'_1 are region-equivalent if for all run prefixes $r[0..k]$, and for all regions R and player-1 actions $a_1 \in A_1^\perp$, we have (1) $\mathcal{W}(\pi_1, r[0..k], R, a_1) = \emptyset$ iff $\mathcal{W}(\pi'_1, r[0..k], R, a_1) = \emptyset$; and (2) $P_{\pi_1}^{r[0..k]}(\mathcal{W}(\pi_1, r[0..k], R, a_1)) = P_{\pi'_1}^{r[0..k]}(\mathcal{W}(\pi'_1, r[0..k], R, a_1))$.

2.3 Winning Sets and Winning Strategies for Timed Automaton Games

In this Subsection we present the computation of winning sets for timed automaton games based on the framework of [dAFH⁺03], and derive various basic properties of winning strategies.

Encoding Time-Divergence by Enlarging the Game Structure. Given a timed automaton game \mathcal{T} , consider the enlarged game structure $\hat{\mathcal{T}}$ (based mostly on the construction in [dAFH⁺03]) with the state space $S^{\hat{\mathcal{T}}} \subseteq S \times \mathbb{R}_{[0,1)} \times \{\text{TRUE}, \text{FALSE}\}^2$, and an augmented transition relation $\delta^{\hat{\mathcal{T}}} : S^{\hat{\mathcal{T}}} \times (M_1 \cup M_2) \mapsto S^{\hat{\mathcal{T}}}$. In an augmented state $\langle s, \mathfrak{z}, \text{tick}, \text{bl}_1 \rangle \in S^{\hat{\mathcal{T}}}$, the component $s \in S$ is a state of the original game structure $\llbracket \mathcal{T} \rrbracket$, \mathfrak{z} is value of a fictitious clock z which gets reset to 0 every time it crosses 1 (i.e., if κ' is the clock valuation resulting from letting time Δ elapse from an initial clock valuation κ , then, $\kappa'(z) = (\kappa(z) + \Delta) \bmod 1$), tick is true iff z crossed 1 at last transition and bl_1 is true if player 1 is to blame for the last transition (ie., blame_1 is true for the last transition). Note that any strategy π_i in $\llbracket \mathcal{T} \rrbracket$, can be considered a strategy in $\hat{\mathcal{T}}$. The values of the clock z , tick and bl_1 correspond to the values each player keeps in memory in constructing his strategy. Given any initial value of $\mathfrak{z} = \mathfrak{z}^*, \text{tick} = \text{tick}^*, \text{bl}_1 = \text{bl}_1^*$; any run r in \mathcal{T} has a corresponding unique run \hat{r} in $\hat{\mathcal{T}}$ with $\hat{r}[0] = \langle r[0], \mathfrak{z}^*, \text{tick}^*, \text{bl}_1^* \rangle$ such that r is a projection of \hat{r} onto \mathcal{T} . For an objective Φ , we can now encode time-divergence as the objective: $\text{TimeDivBl}_1(\Phi) = (\Box \Diamond \text{tick} \rightarrow \Phi) \wedge (\neg \Box \Diamond \text{tick} \rightarrow \Diamond \Box \neg \text{bl}_1)$, where \Box and \Diamond are the standard LTL modalities (“always” and “eventually” respectively), the combinations $\Box \Diamond$ and $\Diamond \Box$ denoting “infinitely often” and “all but for a finite number of steps” respectively. This is formalized in the following proposition.

Proposition 1 (TimeDivBl₁() in terms of tick, bl₁). *Let \mathcal{T} be a timed automaton game and $\hat{\mathcal{T}}$ be the corresponding enlarged game structure. Let Φ be an objective on \mathcal{T} . Consider a run $r = s^0, \langle m_1^0, m_2^0 \rangle, s^1, \langle m_1^1, m_2^1 \rangle, \dots$ in \mathcal{T} . Let \hat{r} denote the corresponding run in $\hat{\mathcal{T}}$ such that $\hat{r} = \langle s^0, \mathfrak{z}^0, \text{tick}^0, \text{bl}_1^0 \rangle, \langle m_1^0, m_2^0 \rangle, \langle s^1, \mathfrak{z}^1, \text{tick}^1, \text{bl}_1^1 \rangle, \langle m_1^1, m_2^1 \rangle$ with $\mathfrak{z}^0 = 0, \text{tick}^0 = \text{FALSE}, \text{bl}_1^0 = \text{FALSE}$. Then $r \in \text{TimeDivBl}_1(\Phi)$ iff $\hat{r} \in ((\Box \Diamond \text{tick} \rightarrow \Phi) \wedge (\neg \Box \Diamond \text{tick} \rightarrow \Diamond \Box \neg \text{bl}_1))$*

Proof. Time diverges in the run r iff it diverges in the corresponding run \hat{r} . Also, the run r belongs to Blameless_1 iff the run \hat{r} belongs to Blameless_1 , which happens iff player 1 is blamed only finitely often, ie., $\Diamond \Box \neg \text{bl}_1$ holds. Hence $r \in \text{TimeDivBl}_1(\Phi)$ iff $\hat{r} \in \text{TimeDivBl}_1(\Phi)$. The result follows from noting that time diverges iff time crosses integer boundaries infinitely often, which happens iff $\Box \Diamond \text{tick}$ holds. \square

The following lemma states that because of the correspondence between \mathcal{T} and $\hat{\mathcal{T}}$, we can obtain the winning sets of \mathcal{T} by obtaining the winning sets in $\hat{\mathcal{T}}$.

Lemma 1 (Equivalence of winning sets of \mathcal{T} and $\widehat{\mathcal{T}}$). *Let \mathcal{T} be a timed automaton game and $\widehat{\mathcal{T}}$ be the corresponding enlarged game structure. Let Φ be an objective on \mathcal{T} . Given any state s of \mathcal{T} , we have $s \in \text{Sure}_1^{\mathcal{T}}(\text{TimeDivBl}_1(\Phi))$ iff $\langle s, 0, \text{FALSE}, \text{FALSE} \rangle \in \text{Sure}_1^{\widehat{\mathcal{T}}}((\Box \Diamond \text{tick} \rightarrow \Phi) \wedge (\neg \Box \Diamond \text{tick} \rightarrow \Diamond \Box \neg \text{bl}_1))$.*

Proof. Consider a state s of \mathcal{T} , and a corresponding state $\langle s, 0, \text{FALSE}, \text{FALSE} \rangle$ of $\widehat{\mathcal{T}}$. The variables \mathbf{z} , tick and bl_1 only “observe” properties in $\widehat{\mathcal{T}}$, they do not restrict transitions. Thus, given a run r of \mathcal{T} from s , there is a unique run \widehat{r} of $\widehat{\mathcal{T}}$ from $\langle s, 0, \text{FALSE}, \text{FALSE} \rangle$ and vice versa. Similarly, any player- i strategy π_i in \mathcal{T} corresponds to a strategy $\widehat{\pi}_i$ in $\widehat{\mathcal{T}}$; and any strategy $\widehat{\pi}_i$ in $\widehat{\mathcal{T}}$ corresponds to a strategy π_i in \mathcal{T} such that both strategies propose the same moves for corresponding runs. The result then follows from Proposition 1. \square

Let $\widehat{\kappa}$ be a valuation for the clocks in $\widehat{C} = C \cup \{z\}$. A state of $\widehat{\mathcal{T}}$ can then be considered as $\langle \langle l, \widehat{\kappa} \rangle, \text{tick}, \text{bl}_1 \rangle$. We extend the clock equivalence relation to these expanded states: $\langle \langle l, \widehat{\kappa} \rangle, \text{tick}, \text{bl}_1 \rangle \cong \langle \langle l', \widehat{\kappa}' \rangle, \text{tick}', \text{bl}_1' \rangle$ iff $l = l'$, $\text{tick} = \text{tick}'$, $\text{bl}_1 = \text{bl}_1'$ and $\widehat{\kappa} \cong \widehat{\kappa}'$. We let $\langle l, \text{tick}, \text{bl}_1 \rangle$ be the “locations” in $\widehat{\mathcal{T}}$. For every ω -regular location objective Φ of \mathcal{T} , we have $\text{TimeDivBl}(\Phi)$ to be an ω -regular location objective of $\widehat{\mathcal{T}}$.

We start first recall the statement of a classical result of [AD94] that the region equivalence relation induces a time abstract bisimulation on the regions.

Lemma 2 ([AD94]). *Let Y, Y' be regions in the timed game structure \mathcal{T} . Suppose player i has a move from $s_1 \in Y$ to $s'_1 \in Y'$, for $i \in \{1, 2\}$. Then, for any $s_2 \in Y$, player i has a move from s_2 to some $s'_2 \in Y'$.*

Let Y, Y'_1, Y'_2 be regions. We prove in Lemma 3 that one of the following two conditions hold: (a) for all states in Y there is a move for player 1 with destination in Y'_1 , such that against all player 2 moves with destination in Y'_2 , the next state is guaranteed to be in Y'_1 ; or (b) for all states in Y for all moves for player 1 with destination in Y'_1 there is a move of player 2 to ensure that the next state is in Y'_2 ; or (c) if $Y'_1 = Y'_2$ (except for the bl_1 component), then player 2 can pick the same time delay as player 1 and hence the winning move is decided by the scheduler. The proof of the lemma is in the appendix.

Lemma 3 (Regions suffice for determining winning move). *Let \mathcal{T} be a timed automaton game, and let Y, Y'_1, Y'_2 be regions in the corresponding enlarged timed game structure $\widehat{\mathcal{T}}$. Suppose player- i has a move $\langle \Delta_i, \perp_i \rangle$ from some $\widehat{s} \in Y$ to $\widehat{s}_i \in Y'_i$, for $i \in \{1, 2\}$. Then, for all states $\widehat{s} \in Y$ and for all player-1 moves $m_1^{\widehat{s}} = \langle \Delta_1, a_1 \rangle$ with $\widehat{s} + \Delta_1 \in Y'_1$, one of the following cases must hold.*

1. $Y'_1 \neq Y'_2$ and for all moves $m_2^{\widehat{s}} = \langle \Delta_2, a_2 \rangle$ of player-2 with $\widehat{s} + \Delta_2 \in Y'_2$, we have $\Delta_1 < \Delta_2$ (and hence $\text{blame}_1(\widehat{s}, m_1^{\widehat{s}}, m_2^{\widehat{s}}, \widehat{\delta}(\widehat{s}, m_1^{\widehat{s}})) = \text{TRUE}$ and $\text{blame}_2(\widehat{s}, m_1^{\widehat{s}}, m_2^{\widehat{s}}, \widehat{\delta}(\widehat{s}, m_2^{\widehat{s}})) = \text{FALSE}$).
2. $Y'_1 \neq Y'_2$ and for all player-2 moves $m_2^{\widehat{s}} = \langle \Delta_2, a_2 \rangle$ with $\widehat{s} + \Delta_2 \in Y'_2$, we have $\Delta_2 < \Delta_1$ (and hence $\text{blame}_2(\widehat{s}, m_1^{\widehat{s}}, m_2^{\widehat{s}}, \widehat{\delta}(\widehat{s}, m_2^{\widehat{s}})) = \text{TRUE}$ and $\text{blame}_1(\widehat{s}, m_1^{\widehat{s}}, m_2^{\widehat{s}}, \widehat{\delta}(\widehat{s}, m_1^{\widehat{s}})) = \text{FALSE}$).
3. $Y'_1 = Y'_2$ and there exists a player 2 move $m_2^{\widehat{s}} = \langle \Delta_2, a_2 \rangle$ with $\widehat{s} + \Delta_2 \in Y'_2$ such that $\Delta_1 = \Delta_2$ (and hence $\text{blame}_1(\widehat{s}, m_1^{\widehat{s}}, m_2^{\widehat{s}}, \widehat{\delta}(\widehat{s}, m_1^{\widehat{s}})) = \text{TRUE}$ and $\text{blame}_2(\widehat{s}, m_1^{\widehat{s}}, m_2^{\widehat{s}}, \widehat{\delta}(\widehat{s}, m_2^{\widehat{s}})) = \text{TRUE}$).

We now show that (1) pure strategies of player 1 suffice for winning from Sure_1 states; and (2) pure strategies of player 2 suffice for spoiling from states that are not Sure_1 .

Lemma 4 (Existence of pure strategies for sure winning sets). *Let \mathcal{G} be a timed game structure, and let Φ be an objective of \mathcal{G} .*

1. *Pure strategies of player 1 suffice for winning from $\text{Sure}_1^{\mathcal{G}}(\Phi)$.*
2. *Pure strategies of player 2 suffice for preventing sure winning of player 1 from states outside of $\text{Sure}_1^{\mathcal{G}}(\Phi)$.*

Proof. 1. Let π_1 be a sure-winning player-1 receptive strategy. Consider any player-1 pure receptive strategy π'_1 such that for any run r of \mathcal{G} , we have $\pi'_1(r[0..k]) \in \text{Support}(\pi_1(r[0..k]))$. Since π_1 is sure-winning, π'_1 must be sure winning too.

2. Let $s \notin \text{Sure}_1^g(\Phi)$ and let π_1 be any player-1 receptive strategy. Let π_2 be a player-2 spoiling receptive strategy against π_1 for the state s . We have $\text{Outcomes}(s, \pi_1, \pi_2) \not\subseteq \Phi$. This means there exists a run $r^* = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \dots$ with $m_i^k \in \text{Support}(\pi_i(r^*[0..k]))$ for $i \in \{1, 2\}$ such that $r^* \notin \Phi$. Consider the pure player-2 receptive strategy π'_2 such that

$$\pi'_2(r[0..k]) = \begin{cases} m_2^k & \text{if } r[0..k] = r^*[0..k] \\ \langle \Delta_2, a_2 \rangle & \text{otherwise, with } \langle \Delta_2, a_2 \rangle \text{ being in the support of } \pi_2(r[0..k]) \end{cases}$$

The receptive strategy π'_2 spoils π_1 from winning surely from s as r^* belongs to $\text{Outcomes}(s, \pi_1, \pi'_2)$, and is not in Φ . \square

Lemma 4 gives us the following corollary which states that Sure_1 sets are equal to the winning sets if only pure strategies are allowed for both players.

Corollary 1 (Equivalence of Pure_1 and Sure_1 sets). *Let \mathcal{T} be a timed automaton game and $\hat{\mathcal{T}}$ be the corresponding enlarged game structure. Let $\hat{\Phi}$ be an ω -regular location objective of $\hat{\mathcal{T}}$, and let $\text{Pure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$ denote the winning set for player 1 when both players are restricted to using only pure strategies. Then, $\text{Pure}_1^{\hat{\mathcal{T}}}(\hat{\Phi}) = \text{Sure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$.*

A μ -calculus formulation for describing the sure winning sets. Given an ω -regular objective $\hat{\Phi}$ of the expanded game structure $\hat{\mathcal{T}}$, a μ -calculus formula φ to describe the winning set $\text{Pure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$ (which is equal to $\text{Sure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$ by Corollary 1) is given in [dAFH⁺03]. The μ -calculus formula uses the *controllable predecessor* operator for player 1, $\text{CPre}_1 : 2^{\hat{S}} \mapsto 2^{\hat{S}}$ (where $\hat{S} = S^{\hat{\mathcal{T}}}$), defined formally by $\hat{s} \in \text{CPre}_1(Z)$ iff $\exists m_1 \in \Gamma_1^{\hat{\mathcal{T}}}(\hat{s}) \forall m_2 \in \Gamma_2^{\hat{\mathcal{T}}}(\hat{s}). \delta_{\text{id}}^{\hat{\mathcal{T}}}(\hat{s}, m_1, m_2) \subseteq Z$. Informally, $\text{CPre}_1(Z)$ consists of the set of states from which player 1 can ensure that the next state will be in Z , no matter what player 2 does. The operator CPre_1 preserves regions of $\hat{\mathcal{T}}$ (this follows from the results of Lemma 3). It was also shown in [dAFH⁺03] that only unions of regions arise in the μ -calculus iteration for ω -regular location objectives.

We now present a lemma that pure finite-memory strategies suffice for winning ω -regular objectives, and all strategies region-equivalent to a region winning strategy are also winning.

Lemma 5 (Properties of pure winning strategies). *Let \mathcal{T} be a timed automaton game and $\hat{\mathcal{T}}$ be the corresponding enlarged game structure. Let $\hat{\Phi}$ be an ω -regular location objective of $\hat{\mathcal{T}}$. Then the following assertions hold.*

- If π is a player-1 pure strategy that wins against all player-2 pure strategies from state \hat{s} , then π_1 wins against all player-2 strategies from state \hat{s} .
- There is a pure finite-memory region strategy π_1 that is sure winning for $\hat{\Phi}$ from the states in $\text{Sure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$.
- If π_1 is a pure region strategy that is sure winning for $\hat{\Phi}$ from $\text{Sure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$ and π'_1 is a pure strategy that is region-equivalent to π_1 , then π'_1 is a sure winning strategy for $\hat{\Phi}$ from $\text{Sure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$.

Proof. 1. Since π_1 wins against all player 2 pure strategies, it must also win against all player 2 strategies (possibly randomized) from \hat{s} (a randomized player-2 strategy may be viewed as a random choice over pure player-2 strategies).

2. It follows from the μ -calculus formulation of [dAFH⁺03] that there exists a pure finite-memory region strategy π_1 that wins against any pure player 2 strategy from the states in $\text{Pure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$. From the previous result, π_1 wins against all player 2 strategies (possibly randomized) from $\text{Pure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$. The claim is proved noting that $\text{Pure}_1^{\hat{\mathcal{T}}}(\hat{\Phi}) = \text{Sure}_1^{\hat{\mathcal{T}}}(\hat{\Phi})$ from Corollary 1.

3. Let π_1 be a pure region strategy that is sure winning for $\hat{\Phi}$ from a state \hat{s} . Let π_1^* be a player-1 pure strategy that is region equivalent to π_1 . The strategy π_1^* is a region strategy as π_1 is a region strategy. We show that π_1^* wins against all player-2 pure strategies. The result then follows from the first part of the lemma.

Consider any player-2 pure strategy π_2 . Suppose π_2 spoils the player-1 strategy π_1^* from winning for $\widehat{\Phi}$. Then, there from the state \widehat{s} there exists a run $\widehat{r}^* = \widehat{s}_0, \langle m_1^0, m_2^0 \rangle, \widehat{s}_1, \langle m_1^1, m_2^1 \rangle, \dots$ with $m_1^k = \pi_1^*(\widehat{r}^*[0..k])$ and $m_2^k = \pi_2(\widehat{r}^*[0..k])$ such that $\widehat{r}^* \notin \widehat{\Phi}$. We show that there exists a player-2 pure strategy π_2^\dagger and a run $\widehat{r}^\dagger \in \text{Outcomes}(\widehat{s}, \pi_1, \pi_2^\dagger)$ with $\text{Reg}(\widehat{r}^*) = \text{Reg}(\widehat{r}^\dagger)$ (contradicting the assumption that π_1 was a player-1 winning strategy). Intuitively, the strategy π_2^\dagger prescribes moves to the same regions as π_2 if the region sequence observed is the same as that of $\text{Reg}(\widehat{r}^*)$. Formally, the strategy π_2^\dagger is defined as follows. Given a run \widehat{r} ,

$$\pi_2^\dagger(\widehat{r}[0..k]) = \begin{cases} \langle \Delta_2, a_2 \rangle & \text{if } \text{Reg}(\widehat{r}[0..k]) = \text{Reg}(\widehat{r}^*[0..k]), \text{ and } \pi_1^*(\widehat{r}^*[0..k]) = \langle \Delta_1^*, a_1 \rangle, \text{ and} \\ & \pi_2(\widehat{r}^*[0..k]) = \langle \Delta_2^*, a_2 \rangle, \text{ and } \Delta_1^* \bowtie \Delta_2^* \text{ for } \bowtie \in \{<, >, =\}, \text{ and} \\ & \pi_1(\widehat{r}[0..k]) = \langle \Delta_1, a_1 \rangle \text{ with } \text{Reg}(\widehat{r}[k] + \Delta_1) = \text{Reg}(\widehat{r}^*[k] + \Delta_1^*) \\ & \text{(observe that } \pi_1 \text{ is a region strategy and } \pi_1^* \text{ is region equivalent to } \pi_1), \\ & \text{and } \Delta_2 \text{ is such that } \text{Reg}(\widehat{r}[k] + \Delta_2) = \text{Reg}(\widehat{r}^*[k] + \Delta_2^*) \text{ and } \Delta_1 \bowtie \Delta_2 \\ & \text{(such a } \Delta_2 \text{ must exist by Lemma 3.)} \\ \langle 0, \perp_2 \rangle & \text{otherwise.} \end{cases}$$

It can be checked that there exists a run $\widehat{r}^\dagger \in \text{Outcomes}(\widehat{s}, \pi_1, \pi_2^\dagger)$ such that $\text{Reg}(\widehat{r}^\dagger) = \text{Reg}(\widehat{r}^*)$. This contradicts the fact that π_1 was a winning strategy. Thus, there cannot exist a player-2 pure strategy π_2 which prevents the player-1 strategy π_1^* from winning. Hence, from the first part of the Lemma, π_1^* is a player-1 winning strategy. \square

Note that there is an infinitely precise global clock z in the enlarged game structure $\widehat{\mathcal{T}}$. If \mathcal{T} does not have such a global clock, then strategies in $\widehat{\mathcal{T}}$ correspond to strategies in \mathcal{T} where player 1 (and player 2) maintain the value of the infinitely precise global clock in memory (requiring infinite memory).

3 Pure Finite-memory Receptive Strategies for Safety Objectives

In this section we show the existence of pure finite-memory sure winning strategies for safety objectives in timed automaton games, and their memory requirements. The encoding of time-divergence in Subsection subsection:ResultsTimedAutomatonGames required an infinitely precise which had to be kept in memory of player 1, requiring infinite memory. In this section, we derive an alternative characterization of receptive strategies which does not requires this extra clock. The characterization of receptive strategies is then used to derive receptive strategies for safety objectives. We also show that our derived winning strategies for safety objectives require only $(|C| + 1)$ memory (where C is the set of clocks of the timed automaton game).

3.1 Analyzing Spoiling Strategies of Player 2

In this subsection we analyze the spoiling strategies of player 2. This analysis will be used in characterizing the receptive strategies of player 1.

Adding predicates to the game structure. We add some predicates to timed automaton games; the predicates will be used later to analyze receptive safety strategies. Given a timed automaton game \mathcal{T} and a state s of \mathcal{T} , we define two functions $V_{>0} : C \mapsto \{\text{TRUE}, \text{FALSE}\}$ and $V_{\geq 1} : C \mapsto \{\text{TRUE}, \text{FALSE}\}$. We obtain $2 \cdot |C|$ predicates based on the two functions. For a clock x , the values of the predicates $V_{>0}(x)$ and $V_{\geq 1}(x)$ indicate if the value of clock x was greater than 0, or greater than or equal to 1 respectively, at the transition point, just before the reset map. For example, for a state $s^p = \langle l^p, \kappa^p \rangle$ and $\delta(s^p, \langle \Delta, a_1 \rangle) = s$, the predicate $V_{>0}(x)$ is TRUE at state s iff $\kappa'(x) > 0$ for $\kappa' = \kappa^p + \Delta$. Consider the enlarged game structure $\widetilde{\mathcal{T}}$ with the state space $\widetilde{S} = S \times \{\text{TRUE}, \text{FALSE}\} \times \{\text{TRUE}, \text{FALSE}\}^C \times \{\text{TRUE}, \text{FALSE}\}^C$ and an augmented transition relation $\widetilde{\delta}$. A state of $\widetilde{\mathcal{T}}$ is a tuple $\langle s, bl_1, V_{>0}, V_{\geq 1} \rangle$, where s is a state of \mathcal{T} , the component bl_1 is TRUE iff player 1 is to be blamed for the last transition, and $V_{>0}, V_{\geq 1}$ are as defined earlier. The clock equivalence relation can be lifted to states of $\widetilde{\mathcal{T}} : \langle s, bl_1, V_{>0}, V_{\geq 1} \rangle \cong_{\widetilde{A}} \langle s', bl'_1, V'_{>0}, V'_{\geq 1} \rangle$ iff $s \cong_{\mathcal{T}} s'$, $bl_1 = bl'_1$, $V_{>0} = V'_{>0}$ and $V_{\geq 1} = V'_{\geq 1}$. We next present

a finite state concurrent game $\tilde{\mathcal{T}}^F$ based on the regions of $\tilde{\mathcal{T}}$ which will be used to analyze spoiling strategies of player 2.

Finite state concurrent game $\tilde{\mathcal{T}}^F$ based on the regions of $\tilde{\mathcal{T}}$. We first show that there exists an finite state concurrent game $\tilde{\mathcal{T}}^F$ which can be used to obtain winning sets and winning strategies of $\tilde{\mathcal{T}}$. The two ideas behind $\tilde{\mathcal{T}}^F$ are that (1) only region sequences are important for games with ω -regular location objectives, and (2) only the destination regions of the players are important (due to Lemma 3). Formally, the game $\tilde{\mathcal{T}}^F$ is defined as the tuple $\langle S^F, M_1^F, M_2^F, \Gamma_1^F, \Gamma_2^F, \delta^F \rangle$ where

- S^F is the set of states of $\tilde{\mathcal{T}}^F$, and is equal to the set of regions of $\tilde{\mathcal{T}}$.
- M_i^F for $i \in \{1, 2\}$ is the set of moves of player- i .
 - $M_1^F = \{ \langle \tilde{R}, a_1 \rangle \mid \tilde{R} \text{ is a region of } \tilde{\mathcal{T}}, \text{ and } a_1 \in A_1^\perp \}$.
 - $M_2^F = \{ \langle \tilde{R}, a_2, i \rangle \mid \tilde{R} \text{ is a region of } \tilde{\mathcal{T}}, i \in \{1, 2\}, \text{ and } a_2 \in A_2^\perp \}$.
- Intuitively, the moves of player- i denote which region it wants to let time pass to, and then take the discrete action a_i^\perp . In addition, for player 2, the “ i ” denotes which player’s move will be chosen should the two players propose moves to the same region. Recall from Lemma 3 that in such a case, it is up to the scheduler to decide which player’s move to “win” in a run. Here, the scheduler is collaborating with player 2.
- Γ_i^F for $i \in \{1, 2\}$ is the move assignment function. Given a state $\tilde{R} \in S^F$, we have $\Gamma_i^F(\tilde{R})$ to be the set of moves available to player i at state \tilde{R} .
 - $\Gamma_1^F(\tilde{R}) = \{ \langle \tilde{R}', a_1 \rangle \mid \exists \tilde{s} \in \tilde{R} \text{ such that player 1 has a move } \langle \Delta, a_1 \rangle \text{ in } \tilde{\mathcal{T}} \text{ from } \tilde{s} \text{ with } \text{Reg}(\tilde{s} + \Delta) = \tilde{R}' \}$.
 - $\Gamma_2^F(\tilde{R}) = \{ \langle \tilde{R}', a_2, i \rangle \mid \exists \tilde{s} \in \tilde{R} \text{ such that player 2 has a move } \langle \Delta, a_2 \rangle \text{ in } \tilde{\mathcal{T}} \text{ from } \tilde{s} \text{ with } \text{Reg}(\tilde{s} + \Delta) = \tilde{R}' \text{ and } i \in \{1, 2\} \}$.
- The transition function δ^F is specified as $\delta^F(\tilde{R}, \langle \tilde{R}_1, a_1 \rangle, \langle \tilde{R}_2, a_2, i \rangle) =$

$$\begin{cases} \tilde{R}' & \text{if } \tilde{R}_1 \neq \tilde{R}_2, \tilde{R}_2 \text{ is a time successor of } \tilde{R}_1, \text{ and } \exists \tilde{s}_1 \in \tilde{R}_1 \text{ such that } \delta^{\tilde{\mathcal{T}}}(\tilde{s}_1, \langle 0, a_1 \rangle) \in \tilde{R}' \\ \tilde{R}' & \text{if } \tilde{R}_1 \neq \tilde{R}_2, \tilde{R}_1 \text{ is a time successor of } \tilde{R}_2, \text{ and } \exists \tilde{s}_2 \in \tilde{R}_2 \text{ such that } \delta^{\tilde{\mathcal{T}}}(\tilde{s}_2, \langle 0, a_2 \rangle) \in \tilde{R}' \\ \tilde{R}' & \text{if } \tilde{R}_1 = \tilde{R}_2, i = 1 \text{ and } \exists \tilde{s}_1 \in \tilde{R}_1 \text{ such that } \delta^{\tilde{\mathcal{T}}}(\tilde{s}_1, \langle 0, a_1 \rangle) \in \tilde{R}' \\ \tilde{R}' & \text{if } \tilde{R}_1 = \tilde{R}_2, i = 2 \text{ and } \exists \tilde{s}_2 \in \tilde{R}_2 \text{ such that } \delta^{\tilde{\mathcal{T}}}(\tilde{s}_2, \langle 0, a_2 \rangle) \in \tilde{R}' \end{cases}$$

Note that given player-1 and player-2 pure strategies $\pi_1^{\tilde{\mathcal{T}}^F}$ and $\pi_2^{\tilde{\mathcal{T}}^F}$, and any state \tilde{R} , we have only one run in $\text{Outcomes}(\tilde{R}, \pi_1^{\tilde{\mathcal{T}}^F}, \pi_2^{\tilde{\mathcal{T}}^F})$.

Mapping runs and states in $\tilde{\mathcal{T}}$ to those in $\tilde{\mathcal{T}}^F$ using $\text{RegMap}()$ and $\text{RegStates}()$. Given a run $\tilde{r} = \tilde{s}_0, \langle m_1^0, m_2^0 \rangle, \tilde{s}_1, \langle m_1^1, m_2^1 \rangle, \dots$ of $\tilde{\mathcal{T}}$, we let $\text{RegMap}(\tilde{r})$ be the corresponding run in $\tilde{\mathcal{T}}^F$ such that the states in \tilde{r} are mapped to their regions, and the moves of $\tilde{\mathcal{T}}$ are mapped to corresponding moves in $\tilde{\mathcal{T}}^F$. Formally, $\text{RegMap}(\tilde{r})$ is the run $\text{Reg}(\tilde{s}_0), \langle m_1^{0,F}, m_2^{0,F} \rangle, \text{Reg}(\tilde{s}_1), \langle m_1^{1,F}, m_2^{1,F} \rangle, \dots$ in $\tilde{\mathcal{T}}^F$ such that for $m_1^j = \langle \Delta_1^j, a_1^j \rangle$ and $m_2^j = \langle \Delta_2^j, a_2^j \rangle$ we have (1) $m_1^{j,F} = \langle \text{Reg}(\tilde{s}_j + \Delta_1^j), a_1^j \rangle$, and (2) $m_2^{j,F} = \langle \text{Reg}(\tilde{s}_j + \Delta_2^j), a_2^j, i \rangle$ with $i = 1$ if $\Delta_1^j < \Delta_2^j$, or $\Delta_1^j = \Delta_2^j$ and $\tilde{s}_{j+1} = \delta(\tilde{s}_j, m_1^j)$ (i.e., the scheduler picks player 1 in round j); otherwise $i = 2$. Given a set of regions X of $\tilde{\mathcal{T}}$ (i.e., X is a set of states of $\tilde{\mathcal{T}}^F$), let $\text{RegStates}(X) = \{ \tilde{s} \mid \tilde{s} \in \bigcup X \}$.

We have the following lemma which states the equivalence of the games $\tilde{\mathcal{T}}^F$ and $\tilde{\mathcal{T}}$ with respect to the CPre_1 operator of the μ -calculus formulation mentioned in Section 2.

Lemma 6. *Let \mathcal{T} be a timed automaton game, $\tilde{\mathcal{T}}$ the expanded game structure as mentioned above, and $\tilde{\mathcal{T}}^F$ the corresponding finite state concurrent game structure. If X is a set of regions of $\tilde{\mathcal{T}}$, then $\text{CPre}_1^{\tilde{\mathcal{T}}^F}(\bigcup X) = \text{RegStates}(\text{CPre}_1^{\tilde{\mathcal{T}}}(X))$*

Proof. The proof follows from Lemma 3. □

Lemma 7 (Relating sure winning sets in $\tilde{\mathcal{T}}^F$ and $\tilde{\mathcal{T}}$). *Let \mathcal{T} be a timed automaton game, $\tilde{\mathcal{T}}$ the expanded game structure as described above, and $\tilde{\mathcal{T}}^F$ the corresponding finite state concurrent game structure. Let $\tilde{\Phi}$ be an ω -regular location objective of $\tilde{\mathcal{T}}$ (and naturally also of $\tilde{\mathcal{T}}^F$). We have $\text{Sure}_1^{\tilde{\mathcal{T}}}(\tilde{\Phi}) = \text{RegStates}(\text{Sure}_1^{\tilde{\mathcal{T}}^F}(\tilde{\Phi}))$.*

Proof. Only unions of regions arise in the μ -calculus iteration for computing winning sets in $\tilde{\mathcal{T}}$ for ω -regular objectives. The proof follows from the fact that equivalent sets of states arise in the μ -calculus iteration for computing the winning sets in both game structures due to Lemma 6. Corollary 1 gives us the equivalence between Pure₁ and Sure₁ sets. \square

Obtaining a Class of Spoiling Player-2 Spoiling Strategies in $\tilde{\mathcal{T}}$ Using the Game Structure $\tilde{\mathcal{T}}^F$.

We use the finite state game $\tilde{\mathcal{T}}^F$ to analyze the spoiling strategies of player 2 for any given player-1 strategy π_1 in $\tilde{\mathcal{T}}$. To do this analysis, we (1) map any player-1 strategy π_1 in $\tilde{\mathcal{T}}$ to a corresponding player-1 strategy π_1^F in $\tilde{\mathcal{T}}^F$; and (2) map any player-2 spoiling strategies in $\tilde{\mathcal{T}}^F$ against π_1^F to a class of player-2 spoiling strategies in $\tilde{\mathcal{T}}$, all of which will be spoiling against π_1 .

We first present the next Lemma which states that for every run of $\tilde{\mathcal{T}}^F$, there exists a run of $\tilde{\mathcal{T}}$ that has an equivalent region sequence.

Lemma 8. *Let \mathcal{T} be a timed automaton game, $\tilde{\mathcal{T}}$ the expanded game structure as described previously, and $\tilde{\mathcal{T}}^F$ the corresponding finite state concurrent game structure. For every finite run \tilde{r}^F of $\tilde{\mathcal{T}}^F$, there exists a finite run \tilde{r} of $\tilde{\mathcal{T}}$ such that $\text{RegMap}(\tilde{r}) = \tilde{r}^F$.*

Proof. Let \tilde{r}^F be any given finite run of $\tilde{\mathcal{T}}^F$. We show by induction on the number of steps in \tilde{r}^F that there exists a finite run of \tilde{r} of $\tilde{\mathcal{T}}$ such that $\text{RegMap}(\tilde{r}) = \tilde{r}^F$. Let the inductive hypothesis be true for all runs with at most j steps. Let \tilde{r}^F contain $j + 1$ steps. By inductive hypothesis, there exists a finite run \tilde{r}^* with j steps such that $\text{RegMap}(\tilde{r}^*) = \tilde{r}^F[0..j]$.

Let $\tilde{r}^F[0..j+1] = \tilde{r}^F[0..j], \langle \langle \tilde{R}_1^j, a_1^j \rangle, \langle \tilde{R}_2^j, a_2^j, i \rangle \rangle, \tilde{R}^j$. Since $\text{Reg}(\tilde{r}^*[j]) = \tilde{r}^F[j]$, we have by Lemma 3, and by the construction of $\tilde{\mathcal{T}}^F$ that (1) there exists a player- i move $\langle \Delta_i, a_i^j \rangle$ from $\tilde{r}^*[j]$ such that $\text{Reg}(\tilde{r}^*[j] + \Delta_i^j) = \tilde{R}_i^j$ for $i \in \{1, 2\}$, and (2) for some $\tilde{s}^* \in \delta_{\text{id}}(\tilde{r}^*[j], \langle \Delta_1, a_1^j \rangle \langle \Delta_2, a_2^j \rangle)$, we have $\text{Reg}(\tilde{s}^*) = \tilde{r}^F[j+1]$. Thus, the run \tilde{r}^* can be extended to \tilde{r} by one more step such that \tilde{r} has the desired properties. \square

Mapping player-1 strategies in $\tilde{\mathcal{T}}$ to player-1 strategies in $\tilde{\mathcal{T}}^F$. Let $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$ be the set of finite runs of $\tilde{\mathcal{T}}^F$. A set of finite runs \mathcal{O} of $\tilde{\mathcal{T}}$ is said to *cover* $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$ if for every (finite) run $\tilde{r}^F \in \text{FinRuns}^{\tilde{\mathcal{T}}^F}$, there exists a *unique* finite run $\tilde{r} \in \mathcal{O}$ such that $\text{RegMap}(\tilde{r}) = \tilde{r}^F$. There exists at least one such run-cover \mathcal{O} by Lemma 8. Abusing notation, we let $\mathcal{O}(\tilde{r}^F)$ denote the unique run $\tilde{r} \in \mathcal{O}$ such that $\text{RegMap}(\tilde{r}) = \tilde{r}^F$. Given a player-1 pure strategy π_1 in $\tilde{\mathcal{T}}$, and a run-cover \mathcal{O} of $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$, we obtain the mapped player-1 pure strategy in $\tilde{\mathcal{T}}^F$, denoted, $\mathbb{F}^{\mathcal{O}}(\pi_1)$, as follows.

$$(\mathbb{F}^{\mathcal{O}}(\pi_1))(\tilde{r}^F) = \begin{cases} \langle \tilde{R}, a_1 \rangle & \text{such that } \pi_1(\mathcal{O}(\tilde{r}^F)) = \langle \Delta_1, a_1 \rangle, \text{ and } \text{Reg}(\mathcal{O}(\tilde{r}^F)[k] + \Delta_1) = \tilde{R} \\ & (\text{where } \mathcal{O}(\tilde{r}^F)[k] \text{ is the last state in } \mathcal{O}(\tilde{r}^F)) \end{cases}$$

Intuitively, the strategy $\mathbb{F}^{\mathcal{O}}(\pi_1)$, on the finite run \tilde{r}^F , acts like π_1 on the finite run $\mathcal{O}(\tilde{r})$ (i.e., the move is to the same region, with the same discrete action).

Mapping player-2 pure strategies in $\tilde{\mathcal{T}}^F$ to player-2 pure strategies in $\tilde{\mathcal{T}}$. We now map any given player-2 pure strategy $\pi_2^{\tilde{\mathcal{T}}^F}$ in $\tilde{\mathcal{T}}^F$ to player-2 pure strategies in $\tilde{\mathcal{T}}$. This mapping will depend on a given player-1 pure strategy π_1 in $\tilde{\mathcal{T}}$ (the strategy π_1 will be given as a parameter). Given a player-2 pure strategy $\pi_2^{\tilde{\mathcal{T}}^F}$ in $\tilde{\mathcal{T}}^F$, and a player-1 pure strategy π_1 in $\tilde{\mathcal{T}}$, we define a *set* of player-2 pure strategies in $\tilde{\mathcal{T}}$. The set, denoted as $\mathbb{T}\text{Set}_{\pi_1}(\pi_2^{\tilde{\mathcal{T}}^F})$, is defined as containing all player-2 pure strategies π_2 in $\tilde{\mathcal{T}}$ satisfying the following condition: given

any run prefix $\tilde{r}[0..k]$ in $\tilde{\mathcal{T}}$, with $\pi_1(\tilde{r}[0..k]) = \langle \Delta_1, a_1 \rangle$, the strategy π_2 satisfies Equation 1.

$$\pi_2(\tilde{r}[0..k]) = \begin{cases} \langle \Delta_2, a_2 \rangle & \text{such that } \Delta_2 < \Delta_1 \text{ and } \text{Reg}(\tilde{r}[k] + \Delta_2) = \tilde{R}_2; \text{ if} \\ & (1) \pi_2^{\tilde{\mathcal{T}}^F}(\text{RegMap}(\tilde{r}[0..k])) = \langle \tilde{R}_2, a_2, i \rangle, \text{ and} \\ & (2) \text{Reg}(\tilde{r}[k] + \Delta_1) \text{ is a time successor of } \tilde{R}_2 \\ \langle \Delta_2, a_2 \rangle & \text{such that } \Delta_2 > \Delta_1 \text{ and } \text{Reg}(\tilde{r}[k] + \Delta_2) = \tilde{R}_2; \text{ if} \\ & (1) \pi_2^{\tilde{\mathcal{T}}^F}(\text{RegMap}(\tilde{r}[0..k])) = \langle \tilde{R}_2, a_2, i \rangle, \text{ and} \\ & (2) \tilde{R}_2 \text{ is a time successor of } \text{Reg}(\tilde{r}[k] + \Delta_1) \\ \langle \Delta_2, a_2 \rangle & \text{such that } \Delta_2 \geq \Delta_1 \text{ and } \text{Reg}(\tilde{r}[k] + \Delta_2) = \tilde{R}_2; \text{ if} \\ & (1) \pi_2^{\tilde{\mathcal{T}}^F}(\text{RegMap}(\tilde{r}[0..k])) = \langle \tilde{R}_2, a_2, 1 \rangle, \text{ and} \\ & (2) \text{Reg}(\tilde{r}[k] + \Delta_1) = \tilde{R}_2 \\ \langle \Delta_2, a_2 \rangle & \text{such that } \Delta_2 \leq \Delta_1 \text{ and } \text{Reg}(\tilde{r}[k] + \Delta_2) = \tilde{R}_2; \text{ if} \\ & (1) \pi_2^{\tilde{\mathcal{T}}^F}(\text{RegMap}(\tilde{r}[0..k])) = \langle \tilde{R}_2, a_2, 2 \rangle, \text{ and} \\ & (2) \text{Reg}(\tilde{r}[k] + \Delta_1) = \tilde{R}_2 \end{cases} \quad (1)$$

Intuitively, a strategy π_2 in $\mathbb{TSet}_{\pi_1}(\pi_2^{\tilde{\mathcal{T}}^F})$ picks a move of time duration bigger than that of π_1 if the strategy $\pi_2^{\tilde{\mathcal{T}}^F}$ in $\tilde{\mathcal{T}}^F$ allows a corresponding player-1 move $\langle \text{Reg}(\tilde{r}[k] + \Delta_1), a_1 \rangle$. Otherwise, the strategies π_2 pick a move of shorter duration.

Player-2 spoiling strategies set $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$ in $\tilde{\mathcal{T}}$. Given a player-1 pure strategy π_1 in $\tilde{\mathcal{T}}$ such that π_1 is not a winning player-1 strategy from a state \tilde{s} (for some ω -regular location objective $\tilde{\Phi}$ of $\tilde{\mathcal{T}}$), we now obtain a specific *set* of player-2 spoiling pure strategies in $\tilde{\mathcal{T}}$ against π_1 from \tilde{s} . The set is denoted as $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$, where \mathcal{O} is a runcover of $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$, and $\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1}$ is a given player-2 spoiling pure strategy against $\mathbb{F}^{\mathcal{O}}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ for the same objective $\tilde{\Phi}$, for the starting state $\text{Reg}(\tilde{s})$. We observe that some player-2 spoiling pure strategy $\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1}$ must exist by Lemma 7 and Corollary 1. The set $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$ of player-2 spoiling pure strategies for π_1 is defined to be equal to $\mathbb{TSet}_{\pi_1}(\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$.

The next Lemma relates spoiling player-2 strategies in $\tilde{\mathcal{T}}^F$ and $\tilde{\mathcal{T}}$ (the proof is by an involved induction argument). The intuition behind the Lemma is that given a state $\tilde{s} \notin \text{win}_1^{\tilde{\mathcal{T}}}(\tilde{\Phi})$, we have that (a) $\text{Reg}(\tilde{s}) \notin \text{win}_1^{\tilde{\mathcal{T}}^F}(\tilde{\Phi})$; and (b) player-2 can obtain spoiling strategies for any player-1 strategy π_1 in $\tilde{\mathcal{T}}$ by prescribing moves to the same regions as the player-2 spoiling strategy in $\tilde{\mathcal{T}}^F$, which spoils $\mathbb{F}^{\mathcal{O}}(\pi_1)$ (for some suitably chosen \mathcal{O}). This result will be used in the next subsection to show that receptive player-1 strategies *must* satisfy certain requirements.

Lemma 9 (Relating spoiling player-2 pure strategies in $\tilde{\mathcal{T}}^F$ and $\tilde{\mathcal{T}}$). *Let \mathcal{T} be a timed automaton game, $\tilde{\mathcal{T}}$ the expanded game structure, and $\tilde{\mathcal{T}}^F$ the corresponding finite state concurrent game structure. Given an ω -regular location objective $\tilde{\Phi}$ of player 1 (in $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}^F$), the following assertions hold.*

1. $\tilde{s} \in \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}}(\tilde{\Phi})$ iff $\text{Reg}(\tilde{s}) \in \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}^F}(\tilde{\Phi})$.
2. Let $\tilde{s} \notin \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}}(\tilde{\Phi})$. Given any player-1 strategy π_1 in $\tilde{\mathcal{T}}$ there exists a runcover \mathcal{O} of $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$ such that for any player-2 pure spoiling strategy $\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1}$ against $\mathbb{F}^{\mathcal{O}}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from the state $\text{Reg}(\tilde{s})$ for the objective $\tilde{\Phi}$ (such spoiling strategies exist by the previous part of the lemma); we have that every player-2 strategy in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$ is a spoiling strategy against π_1 in the structure $\tilde{\mathcal{T}}$ for the objective $\tilde{\Phi}$ from the state \tilde{s} .

Proof. 1. Only unions of regions arise in the μ -calculus iteration to obtain winning sets of player 1 for the objective $\tilde{\Phi}$ in the game structure $\tilde{\mathcal{T}}$. Using Lemma 6 in the μ -calculus iteration for obtaining the player-1 winning set for $\tilde{\Phi}$, we deduce that $\tilde{s} \in \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}}(\tilde{\Phi})$ iff $\text{Reg}(\tilde{s}) \in \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}^F}(\tilde{\Phi})$.

2. By the first part of the lemma, we have that $\text{Reg}(\tilde{s}) \notin \text{Pure}_1^{\tilde{\mathcal{T}}^F}(\tilde{\Phi})$. Thus, given any runcover \mathcal{O} , there exists a pure player-2 spoiling strategy $\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1}$ against $\mathbb{F}^{\mathcal{O}}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from the state $\text{Reg}(\tilde{s})$ for the objective $\tilde{\Phi}$. We show that there exists a runcover \mathcal{O} of $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$ such that given any pure player-2 strategy $\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1}$ which spoils $\mathbb{F}^{\mathcal{O}}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from winning the objective $\tilde{\Phi}$ starting from $\text{Reg}(\tilde{s})$, and given any player-2 strategy π_2 from $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$ in $\tilde{\mathcal{T}}$, there exists a run $\tilde{r}^* \in \text{Outcomes}(\tilde{s}, \pi_1, \pi_2)$ in $\tilde{\mathcal{T}}$, such that the region sequence of \tilde{r}^* is the same as the sequence of regions in the (only) run from $\text{Outcomes}(\text{Reg}(\tilde{s}), \mathbb{F}^{\mathcal{O}}(\pi_1), \pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1})$. This proves the Lemma due to the following: since $\pi_2^{\tilde{\mathcal{T}}^F, \mathcal{O}, \pi_1}$ is a player-2 spoiling strategy against $\mathbb{F}^{\mathcal{O}}(\pi_1)$, we must have that $\text{Reg}(\tilde{r}^*)$ satisfies $\neg\tilde{\Phi}$, and hence \tilde{r}^* satisfies $\neg\tilde{\Phi}$ implying π_2 to be a spoiling strategy of player 2 in $\tilde{\mathcal{T}}$ against π_1 . The proof of the statement is by an involved induction. \square

3.2 Characterizing Receptive Strategies Without Using Extra Clocks

We now present characterizations of receptive strategies in timed automaton games, and show that receptiveness can be expressed as an LTL condition on the states of $\tilde{\mathcal{T}}$, from which it follows that receptive strategies require finite memory in timed automaton games. First, we consider the case where all clocks are bounded in the game (i.e., location invariants of the form $\bigwedge_{x \in C} x \leq d_x$ can be put on all locations).

Lemma 10 (Receptive strategies when all clocks bounded in \mathcal{T}). *Let \mathcal{T} be a timed automaton game in which all clocks are bounded (i.e., for all clocks x we have $x \leq d_x$, for constants d_x in all reachable states). Let $\tilde{\mathcal{T}}$ be the enlarged game structure obtained from \mathcal{T} . Then player 1 has a receptive strategy from a state s of \mathcal{T} iff $\langle s, \cdot \rangle \in \text{Sure}_1^{\tilde{\mathcal{T}}}(\Phi)$, where*

$$\Phi = \Box \Diamond (bl_1 = \text{TRUE}) \rightarrow \left(\left(\bigwedge_{x \in C} \Box \Diamond (x = 0) \right) \wedge \left(\begin{array}{c} \Box \Diamond (bl_1 = \text{TRUE}) \wedge \bigwedge_{x \in C} (V_{>0}(x) = \text{TRUE}) \\ \vee \\ \Box \Diamond (bl_1 = \text{FALSE}) \wedge \bigvee_{x \in C} (V_{\geq 1}(x) = \text{TRUE}) \end{array} \right) \right).$$

Proof. We prove inclusion in both directions.

- (\Leftarrow). For a state $\tilde{s} \in \text{Sure}_1^{\tilde{\mathcal{T}}}(\Phi)$, we show that player 1 has a receptive strategy from \tilde{s} . Let π_1 be a pure sure winning region strategy: since Φ is an ω -regular region objective such a strategy exists by Lemma 5. Consider a strategy π'_1 for player 1 that is region-equivalent to π_1 such that whenever the strategy π_1 proposes a move $\langle \Delta, a_1 \rangle$ for any run prefix $\tilde{r}[0..k]$ with $\tilde{r}[k] + \Delta$ satisfying $\bigwedge_{x \in C} (x > 0)$, then π'_1 proposes the move $\langle \Delta', a_1 \rangle$ for $\tilde{r}[0..k]$ such that $\text{Reg}(\tilde{r}[k] + \Delta) = \text{Reg}(\tilde{r}[k] + \Delta')$ and $\tilde{r}[k] + \Delta'$ satisfies $(\bigvee_{y \in C} y > 1/2) \wedge \bigwedge_{x \in C} (x > 0)$. Such a move always exists; in particular, for any state \tilde{s} , if there exists Δ such that $\tilde{s} + \Delta \in R \subseteq \bigwedge_{x \in C} (x > 0)$, then there exists Δ' such that $\tilde{s} + \Delta' \in R \cap ((\bigvee_{y \in C} y > 1/2) \wedge \bigwedge_{x \in C} (x > 0))$. Intuitively, player 1 jumps near the boundary of R . By Lemma 5, π'_1 is also sure-winning for Φ . The strategy π'_1 ensures that in all resulting runs, if player 1 is not blameless, then all clocks are 0 infinitely often (since for all clocks $\Box \Diamond (x = 0)$), and that some clock has value more than 1/2 infinitely often (either due to player 1 ensuring some clock being greater than 1/2 infinitely often; or player 2 playing moves which result in some clock being greater than 1 infinitely often).. This implies time divergence. Hence player 1 has a receptive winning strategy from \tilde{s} .
- (\Rightarrow). For a state $\tilde{s} \notin \text{Sure}_1^{\tilde{\mathcal{T}}}(\Phi)$, we show that player 1 does not have any receptive strategy starting from state \tilde{s} . We have $\neg\Phi \equiv (\Box \Diamond (bl_1 = \text{TRUE})) \wedge (\neg\Psi_1 \vee (\neg\Psi_2 \wedge \neg\Psi_3))$, where

$$\begin{aligned} \neg\Psi_1^\dagger &= \bigvee_{x \in C} \Diamond \Box (x > 0) \\ \neg\Psi_2^\dagger &= \Diamond \Box \left((bl_1 = \text{TRUE}) \rightarrow \left(\bigvee_{x \in C} (V_{>0}(x) = \text{FALSE}) \right) \right) \\ \neg\Psi_3^\dagger &= \Diamond \Box \left((bl_1 = \text{FALSE}) \rightarrow \left(\bigwedge_{x \in C} (V_{\geq 1}(x) = \text{FALSE}) \right) \right) \end{aligned}$$

Recall the finite state game $\tilde{\mathcal{T}}^F$ based on the regions of $\tilde{\mathcal{T}}$. Suppose $\tilde{s} \notin \underline{\text{Sure}}_1^{\tilde{\mathcal{T}}}(\Phi)$. Then $\tilde{s} \notin \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}}(\Phi)$ by Corollary 1. Consider any pure player-1 strategy π_1 in $\tilde{\mathcal{T}}$. By Lemma 9, $\text{Reg}(\tilde{s}) \notin \underline{\text{Pure}}_1^{\tilde{\mathcal{T}}^F}(\Phi)$, and there exists a runcover \mathcal{O} for $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$ such that for any player-2 pure spoiling strategy $\pi_2^{\tilde{\mathcal{T}}^F}$ against $\mathbb{F}^{\mathcal{O}}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from $\text{Reg}(\tilde{s})$, we have that every player-2 strategy in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$ is a spoiling strategy against π_1 in the structure $\tilde{\mathcal{T}}$.

Let \mathcal{O} be such a runcover, and let $\pi_2^{\tilde{\mathcal{T}}^F}$ be any such player-2 strategy against $\mathbb{F}^{\mathcal{O}}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from $\text{Reg}(\tilde{s})$. We show that with an appropriately chosen π_2 in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$, player 2 can ensure that in one of the resulting runs, player 1 is not blameless, and time converges, and hence player 1 does not have a receptive pure strategy in $\tilde{\mathcal{T}}$. The result follows from observing that if player 1 does not have a pure receptive strategy, then it does not have a (possibly randomized) receptive strategy (as a randomized strategy may be viewed as a random choice over pure strategies).

Consider runs $\tilde{r} \in \text{Outcomes}(\tilde{s}, \pi_1, \pi_2)$ for $\pi_2 \in \text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$. One of the runs must satisfy $\neg\Phi$, which can happen in one of the following ways.

- (a) $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_1$. The condition $\neg\Psi_1^\dagger$ means that there is some clock x which eventually stays strictly greater than 0. Since all clocks are bounded, this condition means that the run is time convergent, and player 1 is not blameless.
- (b) $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_2 \wedge \neg\Psi_3$. The clause $\neg\Psi_2$ means that eventually if an action of player 1 is chosen, then for some clock x , the value of x stays at 0 throughout the move (which means that the move of player-1 is of duration 0). This clause $\neg\Psi_3$ means that eventually if an action of player 2 is chosen, then for every clock x , the value of x is strictly less than 1 during the move.

Player 2 can have a strategy which takes moves smaller than $1/2^j$ during the j -th visit to a region \tilde{R} in which every clock x has value less than 1. We formalize the above statement. The strategy $\pi_2^{\tilde{\mathcal{T}}^F}$ spoils $\mathbb{F}^{\mathcal{O}}(\pi_1)$ from winning in $\tilde{\mathcal{T}}^F$ for the objective Φ . Given a run prefix $\tilde{r}[0..k]$ of $\tilde{\mathcal{T}}$, let $\pi_1(\tilde{r}[0..k]) = \langle \Delta_1, a_1 \rangle$. Consider a player-2 strategy π_2 in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$, and let $\pi_2^{\tilde{\mathcal{T}}^F}(\text{RegMap}(\tilde{r}[0..k])) = \langle \tilde{R}_2, a_2, i \rangle$. Let π_2 be a strategy in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$ such that for $\pi_2(\tilde{r}[0..k]) = \langle \Delta_2, a_2 \rangle$ we have $\Delta_2 \leq \Delta_1$ and $\Delta_2 < 1/2^k$ whenever the following conditions hold.

- i. For every clock x , the value of x is strictly less than 1 in \tilde{R}_2 .
- ii. Either
 - A. \tilde{R}_2 is a region predecessor of $\text{Reg}(\tilde{r}[k] + \Delta_1)$; or
 - B. $i = 2$ and $\text{Reg}(\tilde{r}[k] + \Delta_1) = \tilde{R}_2$.

It can be observed from Equation 1 that such a Δ_2 and such a strategy π_2 in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$ always exist. The above condition ensures that if a move of player 2 is chosen to a region \tilde{R} in which every clock x has value less than 1, then the moves are smaller than $1/2^j$ during the j -th stage of the game. The strategy π_2 is a spoiling strategy against π_1 by Lemma 9 as π_2 is in $\text{Spoil}^{\mathcal{O}}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$. Moreover, this strategy ensures that at least one of the resulting runs \tilde{r} satisfies $\neg\Phi$.

- i. If \tilde{r} satisfies $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_1$, then the run is time convergent, and player 1 is not blameless.
- ii. If \tilde{r} satisfies $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_2 \wedge \neg\Psi_3$, then we have that:
 - A. Eventually, every chosen move of player 2 results in a region \tilde{R} in which every clock x has value less than 1, with the duration of the player-2 move being smaller than $1/2^j$ during the j -th stage of the game; and
 - B. Eventually every chosen move of player 1 is of time duration 0.

Thus, time is convergent in the run \tilde{r} and player 1 is not blameless.

Hence, player 1 does not have a pure receptive strategy from \tilde{s} (from which it follows that it does not have any receptive strategy from \tilde{s}). \square

We next present a couple of examples to demonstrate the role of the various clauses in the the formula Φ of Lemma 10.

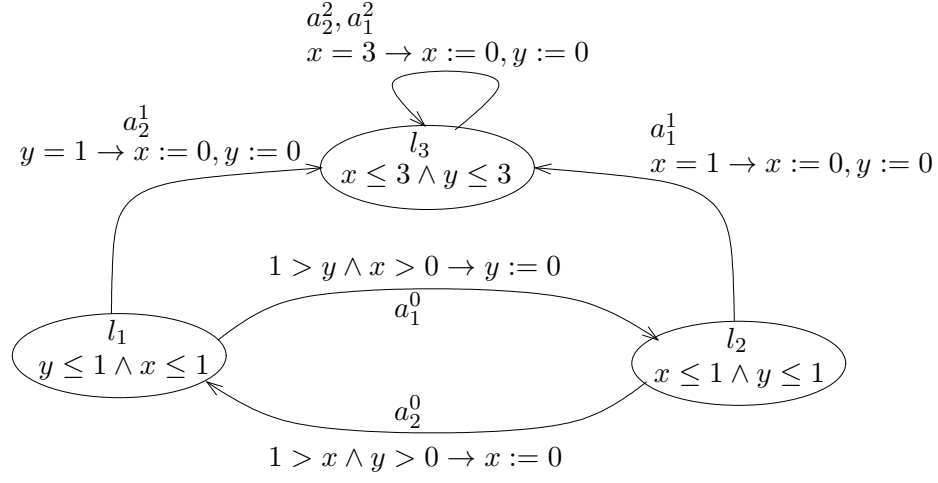


Fig. 2. A time automaton game \mathcal{T}_1 with player-1 receptive strategies.

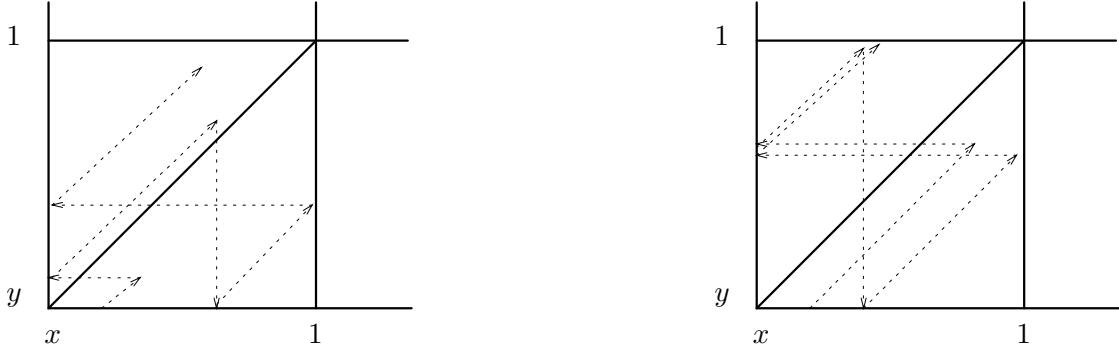


Fig. 3. Two trajectories of the cycle (a_1^0, a_2^0) traversing through two regions of \mathcal{T}_1 .

Example 1. Consider the timed automaton game in Figure 2. The edges a_1^j are player-1 edges and a_2^j player-2 edges. The edges a_2^2 and a_1^2 have the same guards and reset maps. It is clear that player 1 has a receptive strategy when at location l_3 ; it repeatedly takes (or tries to take) the edge a_1^2 . Let us hence focus our attention on plays which consist of (l_1, l_2) cycles (i.e., player 2 picks the edge a_2^0 from location l_2 , and allows player 1 to take the edge a_1^0 from location l_1). Let the starting state satisfy $(x < 1) \wedge (y < 1)$. In a run which consists of (l_1, l_2) cycles, we have that (1) both clocks are reset infinitely often, and (2) both clocks are greater than 0 infinitely often when the edge a_1^0 is taken (this is because the condition on the edge a_2^0 ensures that clock y is greater than 0 when at location l_1 , and the edge condition on a_1^0 further ensure $x > 0$ when edge a_1^0 is taken). Thus, a run of (l_1, l_2) cycles satisfies the formula Φ of Lemma 10. We next illustrate why such a run would be time-divergent (with appropriate chosen player-1 moves for the edge a_1^0).

Observe that after one (l_1, l_2) cycle, the states always satisfy $1 > x > y > 0$ when at l_1 , and $1 > y > x > 0$ when at l_2 . Figure 1 illustrates two paths through these two regions after at least one (l_1, l_2) cycle. Note that the transitions into the region $1 > x > y > 0$ are controlled by player 2, and those into $1 > y > x > 0$ controlled by player 1. In the second trajectory, player 1 is *not* able to take transitions which make the clock x more than $1/2$; but it is able to ensure that the clock y is more than $1/2$ infinitely often. Since the clock y is more than $1/2$ infinitely often and is also reset infinitely often, time diverges (we will present a more formal proof of time divergence of the run shortly). It is easy to construct another timed automaton \mathcal{T}_* in which player 1 can only ensure that clock x is more than $1/2$ infinitely often. It can then be seen that the automata \mathcal{T}_1 and \mathcal{T}_* can be “combined” by a player-2 action so that player 1 can only ensure that *some* clock is more than $1/2$ infinitely often; it cannot ensure that any one *particular* clock will satisfy this property. To ensure time

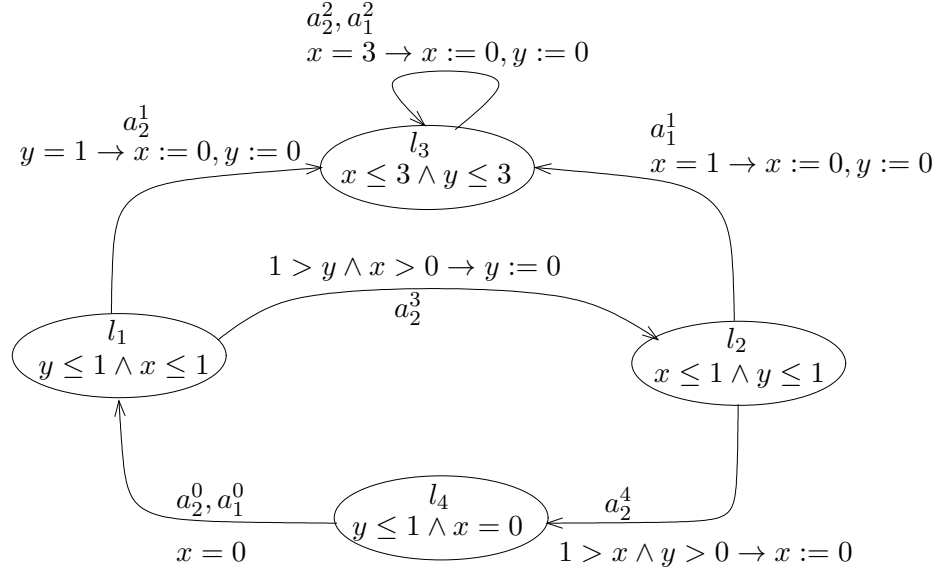


Fig. 4. A timed automaton game \mathcal{T}_2 without player-1 receptive strategies.

divergence, player 1 hence also needs to ensure that *all* clocks are reset infinitely often (as it does not know which clock will be more than $1/2$ infinitely often).

We now formally show time divergence of the runs shown in Figure 1. Let the duration of the j -th player 2 move be Δ_2^j . The value of the clock y is then Δ_2^j when location l_1 is entered for the j -th time, after the j -th a_2^0 move. Player 1 picks its j -th a_1^0 move to be of duration $1 - \Delta_2^j + \varepsilon$. Thus, in one cycle time passes by $1 - \varepsilon$ time units. With $\varepsilon < 1$, it can be seen that time diverges. \square

Example 2. In this example we illustrate why we require in the formula Φ of Lemma 10 that if $\Box\Diamond(bl_1 = \text{TRUE}) \wedge \bigwedge_{x \in C}(V_{>0}(x) = \text{TRUE})$ does not hold, then $\Box\Diamond(bl_1 = \text{FALSE}) \wedge \bigvee_{x \in C}(V_{\geq 1}(x) = \text{TRUE})$ must hold. Consider the timed automaton game \mathcal{T}_2 in Figure 4. The edges a_1^j are player-1 edges and a_2^j player-2 edges. The edges a_2^2 and a_1^2 have the same guards and reset maps. It is clear that player 1 has a receptive strategy when at location l_3 ; it repeatedly takes (or tries to take) the edge a_1^2 . Hence, player 2 keeps the game in the (l_1, l_2, l_4) . For the j -th a_2^3 and the j -th a_2^4 move, player 2 chooses a time duration of $1/2^j$. Player 1 is forced to take the move a_1^0 (of time duration 0) when at location l_4 . In this cycle with such a strategy by player 2, we have that (1) all clocks are reset infinitely often, (2) the moves of player 1 are picked infinitely often, and (3) all clock values are greater than 0 infinitely often (i.e., $\Box\Diamond \bigwedge_{x \in C}(V_{>0}(x) = \text{TRUE})$ holds). But, time converges in such a run (and thus player 1 does not have a receptive strategy). The states in l_1, l_2, l_4 (with $x < 1 \wedge y < 1$) do *not* satisfy Φ of Lemma 10 because even though $\Box\Diamond \bigwedge_{x \in C}(V_{>0}(x) = \text{TRUE})$ holds, $\Box\Diamond(bl_1 = \text{TRUE}) \wedge \bigwedge_{x \in C}(V_{>0}(x) = \text{TRUE})$ does not hold. As this example shows, if player 2 picks moves to satisfy $\bigwedge_{x \in C}(V_{>0}(x) = \text{TRUE})$, then it can choose arbitrarily small moves. That is why require that if we are considering player 2 moves, then $\bigvee_{x \in C}(V_{\geq 1}(x) = \text{TRUE})$ must hold infinitely often. \square

Characterization of receptive strategies for general timed automaton games([CHP08]). Lemma 10 was generalized to all timed automaton games in the following lemma presented in [CHP08]. The idea of the generalization is to identify the subset of clocks which “escape” to infinity; and then to take a disjunction over all such possible subsets. Note that once a clock x becomes more than c_x , then its actual value can be considered irrelevant in determining regions. If only the clocks in $X \subseteq C$ have escaped beyond their maximum tracked values, the rest of the clocks still need to be tracked.

Lemma 11 ([CHP08]). *Let \mathcal{T} be a timed automaton game, and $\tilde{\mathcal{T}}$ be the corresponding enlarged game. Then player 1 has a receptive strategy from a state s iff $\langle s, \cdot \rangle \in \text{Sure}_1^{\tilde{\mathcal{T}}}(\Phi^*)$, where $\Phi^* = \Box\Diamond(bl_1 = \text{TRUE}) \rightarrow \bigvee_{X \subseteq C} \phi_X$,*

and $\phi_X =$

$$\left(\bigwedge_{x \in X} \Diamond \Box (x > c_x) \right) \wedge \left(\left(\bigwedge_{x \in C \setminus X} \Box \Diamond (x = 0) \right) \wedge \left(\begin{array}{c} \Box \Diamond \left((bl_1 = \text{TRUE}) \wedge \bigwedge_{x \in C \setminus X} (V_{>0}(x) = \text{TRUE}) \right) \\ \vee \\ \Box \Diamond \left((bl_1 = \text{TRUE}) \wedge \bigvee_{x \in C \setminus X} (V_{\geq 1}(x) = \text{TRUE}) \right) \end{array} \right) \right)$$

New characterization of receptive strategies for general timed automaton games. We shall see later that player-1 strategies which win for the objective Φ^* of Lemma 11 have a bound of $(|C| + 1)^{2^{|C|}}$ for the number of memory states required. We present a new characterization of receptive strategies for which we can prove a memory bound of only $(|C| + 1)$. First, we need to add $|C|$ predicates to the game structure $\tilde{\mathcal{T}}$. For a state s of \mathcal{T} , we define another function $V_{>\max}^* : C \mapsto \{\text{TRUE}, \text{FALSE}\}$. The value of the predicate $V_{>\max}^*(x)$ for a clock $x \in C$ is TRUE at a state s iff the value of clock x is more than c_x , and was more than c_x in the previous state. That is, if a state $s^p = \langle l^p, \kappa^p \rangle$ and $\delta(s^p, \langle \Delta, a_1 \rangle) = s$, then at the state s , the predicate $V_{>\max}^*(x)$ is TRUE iff $\kappa'(x) > c_x$ for $\kappa' \in \{\kappa^p + \Delta' \mid 0 \leq \Delta' \leq \Delta\}$. Let $\tilde{\mathcal{T}}$ be the enlarged game structure similar to $\tilde{\mathcal{T}}$ with the state space being enlarged to also have $V_{>\max}^*$ values (in addition to $V_{>0}$ and $V_{\geq 1}$ values): $\tilde{S} = S \times \{\text{TRUE}, \text{FALSE}\} \times \{\text{TRUE}, \text{FALSE}\}^C \times \{\text{TRUE}, \text{FALSE}\}^C \times \{\text{TRUE}, \text{FALSE}\}^C$. A state of $\tilde{\mathcal{T}}$ is a tuple $\langle s, bl_1, V_{>0}, V_{\geq 1}, V_{>\max}^* \rangle$, where s is a state of \mathcal{T} , the component bl_1 is TRUE iff player 1 is to be blamed for the last transition, and $V_{>0}, V_{\geq 1}, V_{>\max}^*$ are as defined earlier. A finite state concurrent game $\tilde{\mathcal{T}}^F$ analogous to $\tilde{\mathcal{T}}^F$ can be constructed, and results analogous to Lemmas 6, 7 and 9 hold for the structures $\tilde{\mathcal{T}}$ and $\tilde{\mathcal{T}}^F$.

First we present the following technical Lemma which will be used later.

Lemma 12. *Let \mathcal{T} be a timed automaton game, and $\tilde{\mathcal{T}}$ be the corresponding enlarged game. A run \tilde{r} in $\tilde{\mathcal{T}}$ satisfies*

$$\bigwedge_{x \in C} (\Box \Diamond (x = 0) \vee \Diamond \Box (V_{>\max}^*(x) = \text{TRUE}))$$

iff it satisfies

$$\bigwedge_{x \in C} \Box \Diamond ((x = 0) \vee (V_{>\max}^*(x) = \text{TRUE})).$$

Proof. We prove inclusion in both directions.

1. (\Rightarrow). Suppose a run \tilde{r} in $\tilde{\mathcal{T}}$ satisfies $\bigwedge_{x \in C} (\Box \Diamond (x = 0) \vee \Diamond \Box (V_{>\max}^*(x) = \text{TRUE}))$. Consider a clock $x \in C$. If either $\Box \Diamond (x = 0)$ or $\Diamond \Box (V_{>\max}^*(x) = \text{TRUE})$ holds on \tilde{r} , it can be seen that $\Box \Diamond ((x = 0) \vee (V_{>\max}^*(x) = \text{TRUE}))$ holds on \tilde{r} .
2. (\Leftarrow). Suppose a run \tilde{r} in $\tilde{\mathcal{T}}$ satisfies $\bigwedge_{x \in C} \Box \Diamond ((x = 0) \vee (V_{>\max}^*(x) = \text{TRUE}))$. Consider a clock $x \in C$. We must have either $\Box \Diamond (x = 0)$ or $\Box \Diamond (V_{>\max}^*(x) = \text{TRUE})$. If $\Box \Diamond (x = 0)$ on the run \tilde{r} , then it satisfies our requirement. We show that if run \tilde{r} satisfies $\Box \Diamond (V_{>\max}^*(x) = \text{TRUE})$; then it must satisfy either $\Box \Diamond (x = 0)$ or $\Diamond \Box (V_{>\max}^*(x) = \text{TRUE})$. This is because the only way for the value of a clock to decrease is to be reset to 0. In particular, once the clock x becomes more than c_x , the only way for it to become less than or equal to c_x is to be reset to 0. If the clock x becomes more than c_x and is never reset, it will stay more than c_x forever. \square

Lemma 13 (Receptive strategies when clocks may be unbounded in \mathcal{T}). *Let \mathcal{T} be a timed automaton game, and $\tilde{\mathcal{T}}$ be the corresponding enlarged game. Then player 1 has a receptive strategy from a state s of \mathcal{T} iff*

$\langle s, \cdot \rangle \in \underline{\text{Sure}}_1^{\ddot{\gamma}}(\Phi^\dagger)$, where $\Phi^\dagger = \Box \Diamond (bl_1 = \text{TRUE}) \rightarrow \Psi^\dagger$, and $\Psi^\dagger =$

$$\left(\begin{array}{c} \left(\bigwedge_{x \in C} \Box \Diamond ((x = 0) \vee (V_{>\max}^*(x) = \text{TRUE})) \right) \\ \wedge \\ \left(\Box \Diamond \left((bl_1 = \text{TRUE}) \wedge \left(\bigwedge_{x \in C} (V_{>0}(x) = \text{TRUE}) \right) \wedge \left(\bigvee_{x \in C} (V_{>\max}^*(x) = \text{FALSE}) \right) \right) \right) \\ \vee \\ \left(\Box \Diamond \left((bl_1 = \text{FALSE}) \wedge \bigvee_{x \in C} ((V_{\geq 1}(x) = \text{TRUE}) \wedge (V_{>\max}^*(x) = \text{FALSE})) \right) \right) \end{array} \right) \\ \vee \\ \left(\bigwedge_{x \in C} \Diamond \Box (V_{>\max}^*(x) = \text{TRUE}) \right)$$

Proof. We prove inclusion in both directions.

1. (\Leftarrow). For a state $\ddot{s} \in \underline{\text{Sure}}_1^{\ddot{\gamma}}(\Phi^\dagger)$, we show that player 1 has a receptive strategy from \ddot{s} . Let π_1 be a pure sure winning region strategy: since Φ^\dagger is an ω -regular region objective such a strategy exists by Lemma 5. Let \ddot{R}_{\max} denote the region where for every clock x , the value of x is more than c_x . Consider a region strategy π'_1 for player 1 that is region-equivalent to π_1 such that given a run prefix $\ddot{r}[0..k]$, the strategy π'_1 acts like π_1 except when:
 - If $\text{Reg}(\ddot{r}[k]) = \ddot{R}_{\max}$ and $\pi_1(\ddot{r}[0..k]) = \langle \Delta, a_1 \rangle$, then $\pi'_1(\ddot{r}[0..k]) = \langle \Delta', a_1 \rangle$ such that $\Delta' > 1$ (observe that $\text{Reg}(\ddot{r}[k] + \Delta') = \ddot{R}_{\max}$ for any Δ').
 - If $\text{Reg}(\ddot{r}[k]) \neq \ddot{R}_{\max}$ and $\pi_1(\ddot{r}[0..k]) = \langle \Delta, a_1 \rangle$ with the state $\ddot{r}[k] + \Delta$ being such that the value of some clock x is less than or equal to c_x but more than 0, then $\pi'_1(\ddot{r}[0..k]) = \langle \Delta', a_1 \rangle$ such that (1) $\text{Reg}(\ddot{r}[k] + \Delta') = \text{Reg}(\ddot{r}[k] + \Delta)$, and (2) the value of some clock y (possibly different from x) is less than c_y at $\ddot{r}[k]$, and is more than $1/2$ at $\ddot{r}[k] + \Delta'$ (intuitively, π'_1 jumps near the region boundary of $\text{Reg}(\ddot{r}[k] + \Delta)$).

We have $\Phi^\dagger \equiv (\neg \Box \Diamond (bl_1 = \text{TRUE})) \vee \left(\left(\Psi_1^\dagger \wedge \left(\Psi_2^\dagger \vee \Psi_3^\dagger \right) \right) \vee \Psi_4^\dagger \right)$, where

$$\begin{aligned} \Psi_1^\dagger &= \bigwedge_{x \in C} \Box \Diamond ((x = 0) \vee (V_{>\max}^*(x) = \text{TRUE})) \\ \Psi_2^\dagger &= \Box \Diamond \left((bl_1 = \text{TRUE}) \wedge \left(\bigwedge_{x \in C} (V_{>0}(x) = \text{TRUE}) \right) \wedge \left(\bigvee_{x \in C} (V_{>\max}^*(x) = \text{FALSE}) \right) \right) \\ \Psi_3^\dagger &= \Box \Diamond \left((bl_1 = \text{FALSE}) \wedge \bigvee_{x \in C} ((V_{\geq 1}(x) = \text{TRUE}) \wedge (V_{>\max}^*(x) = \text{FALSE})) \right) \\ \Psi_4^\dagger &= \bigwedge_{x \in C} \Diamond \Box (V_{>\max}^*(x) = \text{TRUE}) \end{aligned}$$

Given any player-2 strategy π_2 , consider any run $\ddot{r} \in \text{Outcomes}(\ddot{s}, \pi'_1, \pi_2)$. The run \ddot{r} must satisfy Φ^\dagger . One of the following conditions must be satisfied on the run \ddot{r} .

- (a) $\Diamond \Box (bl_1 = \text{FALSE})$. This satisfies the receptiveness condition.
- (b) $(\Box \Diamond (bl_1 = \text{TRUE})) \wedge \Psi_4^\dagger$. This means that in the run \ddot{r} , every clock x eventually becomes greater than c_x ; and moves of player 1 are chosen infinitely often. Since the strategy π'_1 chooses moves of duration greater than 1 when staying in \ddot{R}_{\max} , time diverges in the run \ddot{r} .
- (c) $((\Box \Diamond (bl_1 = \text{TRUE})) \wedge \neg \Psi_4^\dagger) \wedge \left(\Psi_1^\dagger \wedge \left(\Psi_2^\dagger \vee \Psi_3^\dagger \right) \right)$. The constraint $\neg \Psi_4^\dagger$ means that, there is some clock x which is less than c_x infinitely often. Satisfaction of the constraint Ψ_1^\dagger and Lemma 12 imply that

$$\bigwedge_{x \in C} (\Box \Diamond (x = 0) \vee \Diamond \Box (V_{>\max}^*(x) = \text{TRUE}))$$

must be satisfied on the run \ddot{r} . That is, each clock x which is not eventually always greater than c_x must be 0 infinitely often. Also, the run must satisfy either Ψ_2^\dagger or Ψ_3^\dagger .

Suppose we have the first case (i.e., Ψ_2^\dagger holds). Then, for infinitely many k , player-1 moves are chosen from $\ddot{r}[0..k]$ such that for some clock x , we have (1) the value of the clock x is less than c_x at $\ddot{r}[k]$ (note that if the value of x is less than c_x at some point during the move, then it must be less than c_x at the origin), and (2) for $\pi'_1(\ddot{r}[0..k]) = \langle \Delta', a_1 \rangle$, the value of the clock x at $\ddot{r}[k] + \Delta'$ is more than 0. Because of the design of π'_1 , this means that for infinitely many k , there is some clock y such that if $\pi'_1(\ddot{r}[0..k]) = \langle \Delta', a_1 \rangle$ then, (1) the value of clock y at $\ddot{r}[k]$ is not more than c_y , and (2) the value of clock y is more than $1/2$ at $\ddot{r}[k] + \Delta'$. Since the clock y must also be equal to 0 infinitely often (as it is not more than c_y eventually from above, and due to Ψ_1^\dagger), this implies that time diverges.

Suppose we have the second case (i.e., Ψ_3^\dagger holds). Then, for infinitely many k , player-2 moves are chosen from $\ddot{r}[0..k]$ such that for some clock x , we have (1) the value of the clock x is less than c_x at $\ddot{r}[k]$, and, (2) for $\pi_2(\ddot{r}[0..k]) = \langle \Delta_2, a_2 \rangle$, the value of the clock x at $\ddot{r}[k] + \Delta_2$ is more than or equal to 1. Since the clock x must also be equal to 0 infinitely often (as it is not more than c_x eventually from above, and due to Ψ_1^\dagger), this implies that time diverges.

Thus, in all cases, the strategy π'_1 ensures that either player 1 is not to blame, or time diverges. Hence, π'_1 is a receptive strategy from \ddot{s} .

2. (\Rightarrow). For a state $\ddot{s} \notin \text{Sure}_1^{\ddot{r}}(\Phi^\dagger)$, we show that player 1 does not have any receptive strategy starting from state \ddot{s} . We have $\neg\Phi^\dagger \equiv (\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\left(\left(\Psi_1^\dagger \wedge \left(\Psi_2^\dagger \vee \Psi_3^\dagger\right)\right) \vee \Psi_4^\dagger\right)$, where $\Psi_1^\dagger, \Psi_2^\dagger, \Psi_3^\dagger$ and Ψ_4^\dagger are as defined previously. Simplifying, we get $\neg\Phi^\dagger \equiv (\Box\Diamond(bl_1 = \text{TRUE})) \wedge \left(\neg\Psi_1^\dagger \vee \left(\neg\Psi_2^\dagger \wedge \neg\Psi_3^\dagger\right)\right) \wedge \neg\Psi_4^\dagger$, where

$$\begin{aligned}\neg\Psi_1^\dagger &= \bigvee_{x \in C} \Diamond\Box((x > 0) \wedge (V_{>\max}^*(x) = \text{FALSE})) \\ \neg\Psi_2^\dagger &= \Diamond\Box\left(\left((bl_1 = \text{TRUE}) \rightarrow \left(\left(\bigvee_{x \in C} ((V_{>0}(x) = \text{FALSE}))\right) \vee \left(\bigwedge_{x \in C} (V_{>\max}^*(x) = \text{TRUE})\right)\right)\right)\right) \\ \neg\Psi_3^\dagger &= \Diamond\Box\left(\left((bl_1 = \text{FALSE}) \rightarrow \bigwedge_{x \in C} ((V_{\geq 1}(x) = \text{FALSE}) \vee (V_{>\max}^*(x) = \text{TRUE}))\right)\right) \\ \neg\Psi_4^\dagger &= \Box\Diamond\bigvee_{x \in C} (V_{>\max}^*(x) = \text{FALSE}) \\ &\quad \text{(Using the identity } \bigvee_{x \in C} \Box\Diamond P(x) \equiv \Box\Diamond \bigvee_{x \in C} P(x)\text{)}\end{aligned}$$

Recall the finite state game $\tilde{\mathcal{T}}^F$ based on the regions of $\tilde{\mathcal{T}}$. There exists a similar finite state game $\tilde{\mathcal{T}}^F$ based on the regions of $\tilde{\mathcal{T}}$, with results relating $\tilde{\mathcal{T}}^F$ and $\tilde{\mathcal{T}}$ as the results relating $\tilde{\mathcal{T}}^F$ and $\tilde{\mathcal{T}}$. Suppose $\ddot{s} \notin \text{Sure}_1^{\ddot{r}}(\Phi^\dagger)$. Then $\ddot{s} \notin \text{Pure}_1^{\ddot{r}}(\Phi^\dagger)$ by Corollary 1. Consider any pure player-1 strategy π_1 in $\tilde{\mathcal{T}}$. By Lemma 9, $\text{Reg}(\ddot{s}) \notin \text{Pure}_1^{\ddot{r}}(\Phi^\dagger)$, and there exists a runcover \mathcal{O} for $\text{FinRuns}^{\tilde{\mathcal{T}}^F}$ such that for any player-2 pure spoiling strategy $\pi_2^{\tilde{\mathcal{T}}^F}$ against $\mathbb{F}^\mathcal{O}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from $\text{Reg}(\ddot{s})$, we have that every player-2 strategy in $\text{Spoil}^\mathcal{O}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$ is a spoiling strategy against π_1 in the structure $\tilde{\mathcal{T}}$.

Let \mathcal{O} be such a runcover, and let $\pi_2^{\tilde{\mathcal{T}}^F}$ be any such player-2 strategy against $\mathbb{F}^\mathcal{O}(\pi_1)$ in $\tilde{\mathcal{T}}^F$ from $\text{Reg}(\ddot{s})$. We show that with an appropriately chosen π_2 in $\text{Spoil}^\mathcal{O}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$, player 2 can ensure that in one of the resulting runs, player 1 is not blameless, and time converges, and hence player 1 does not have a receptive pure strategy in $\tilde{\mathcal{T}}$. The result follows from observing that if player 1 does not have a pure receptive strategy, then it does not have a (possibly randomized) receptive strategy (as a randomized strategy may be viewed as a random choice over pure strategies).

Consider runs $\ddot{r} \in \text{Outcomes}(\ddot{s}, \pi_1, \pi_2)$ for $\pi_2 \in \text{Spoil}^\mathcal{O}(\pi_1, \pi_2^{\tilde{\mathcal{T}}^F})$. One of the runs must satisfy $\neg\Phi^\dagger$, which can happen in one of the following ways.

- (a) $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_1^\dagger \wedge \neg\Psi_4^\dagger$. The condition $\neg\Psi_1^\dagger$ means that there is some clock x which eventually stays strictly greater than 0, and also stays less than or equal to c_x . This is impossible in a time-divergent run as clocks can only be reset to 0. Thus, in this run time does not diverge, and player 1 is not blameless.
- (b) $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_2^\dagger \wedge \neg\Psi_3^\dagger \wedge \neg\Psi_4^\dagger$. The clause $\neg\Psi_4^\dagger$ implies that there is some clock x such that it is not greater than c_x infinitely often during transitions (including the originating state). The clause $\neg\Psi_2^\dagger$ means that eventually if an action of player 1 is chosen, then either (1) every clock x has value greater than c_x during the move (this is not possible if the run satisfies $\neq \Psi_4^\dagger$), or (2) for some clock x , the value of x stays at 0 throughout the move (which means that the move of player-1 is of duration 0). This clause $\neg\Psi_3^\dagger$ means that eventually if an action of player 2 is chosen, then for every clock x , either the clock x has value greater than c_x during the move, or the value of x is strictly less than 1 during the move.

Player 2 can have a strategy which takes moves smaller than $1/2^j$ during the j -th visit to a region \ddot{R} in which every clock x either has value less than 1, or greater than c_x . We formalize the above statement. The strategy $\pi_2^{\ddot{J}^F}$ spoils $\mathbb{F}^O(\pi_1)$ from winning in \ddot{J}^F for the objective Φ^\dagger . Given a run prefix $\ddot{r}[0..k]$ of \ddot{J} , let $\pi_1(\ddot{r}[0..k]) = \langle \Delta_1, a_1 \rangle$. Consider a player-2 strategy π_2 in $\text{Spoil}^O(\pi_1, \pi_2^{\ddot{J}^F})$, and let $\pi_2^{\ddot{J}^F}(\text{RegMap}(\ddot{r}[0..k])) = \langle \ddot{R}_2, a_2, i \rangle$. Let π_2 be a strategy in $\text{Spoil}^O(\pi_1, \pi_2^{\ddot{J}^F})$ such that for $\pi_2(\ddot{r}[0..k]) = \langle \Delta_2, a_2 \rangle$ we have $\Delta_2 \leq \Delta_1$ and $\Delta_2 < 1/2^k$ whenever the following conditions hold.

- i. Each clock x in \ddot{R}_2 is either less than 1, or more than c_x ; and
- ii. Either
 - A. \ddot{R}_2 is a region predecessor of $\text{Reg}(\ddot{r}[k] + \Delta_1)$; or
 - B. $i = 2$ and $\text{Reg}(\ddot{r}[k] + \Delta_1) = \ddot{R}_2$

It can be observed from Equation 1 that such a Δ_2 and such a strategy π_2 in $\text{Spoil}^O(\pi_1, \pi_2^{\ddot{J}^F})$ always exist. The above condition ensures that if a move of player 2 is chosen to a region \ddot{R} in which every clock x either has value less than 1, or greater than c_x , then the moves smaller than $1/2^j$ during the j -th stage of the game. The strategy π_2 is a spoiling strategy against π_1 by Lemma 9 as π_2 is in $\text{Spoil}^O(\pi_1, \pi_2^{\ddot{J}^F})$. Moreover, this strategy ensures that at least one of the resulting runs \ddot{r} satisfies $\neg\Phi^\dagger$.

- i. If \ddot{r} satisfies $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_1^\dagger \wedge \neg\Psi_4^\dagger$, then the run is time convergent, and player 1 is not blameless.
- ii. If \ddot{r} satisfies $(\Box\Diamond(bl_1 = \text{TRUE})) \wedge \neg\Psi_2^\dagger \wedge \neg\Psi_3^\dagger \wedge \neg\Psi_4^\dagger$, then we have that:
 - A. Eventually every chosen move of player 2 results in a region \ddot{R} in which every clock x either has value less than 1, or greater than c_x , with the duration of the player-2 move being smaller than $1/2^j$ during the j -th stage of the game; and
 - B. Eventually every chosen move of player 1 is of time duration 0.

Thus, time is convergent in the run \ddot{r} and player 1 is not blameless.

Hence, in both cases, player 1 does not have a pure receptive strategy from \ddot{s} (from which it follows that it does not have any receptive strategy from \ddot{s}). \square

3.3 Memory Requirement of Receptive Strategies

In this subsection we deduce memory bounds on player-1 receptive strategies using Zielonka tree analysis (see [DJW97] for details). We first deduce a bound that allows player 1 to win in the finite state concurrent game \ddot{J}^F . A player-1 winning strategy in \ddot{J}^F can be mapped to a player-1 winning strategy in \ddot{J} by letting $\pi_1^{\ddot{J}}(\ddot{r}[0..k]) = \langle \Delta, a_1 \rangle$ such that (a) $\pi_1^{\ddot{J}^F}(\text{Reg}(\ddot{r}[0..k])) = \langle \ddot{R}, a_1 \rangle$, and (b) $\text{Reg}(\ddot{r}[k] + \Delta) = \ddot{R}$. Thus, the memory requirement for a player-1 winning strategy in \ddot{J} is not more than as for in the finite game \ddot{J}^F . We note that Zielonka tree analysis holds only for turn based games, but since concurrent games with sure winning conditions reduce to concurrent games in which both players may use only pure strategies, which in turn reduce to turn based games, the Zielonka tree analysis is valid for game \ddot{J}^F with sure winning conditions.

Zielonka tree analysis. Let AP be a set of atomic propositions, and let AP_N be AP together with the negations of the propositions, i.e., $\text{AP} \cup \{\neg P \mid P \in \text{AP}\}$. We say a set $\mathcal{B} \subseteq \text{AP}_N$ is *consistent* with respect to AP iff for

all propositions $P \in \text{AP}$, either $P \in \mathcal{B}$, or $\neg P \in \mathcal{B}$ (or both belong to \mathcal{B}). A *Muller* winning condition \mathcal{F} is a consistent subset of 2^{AP_N} . An infinite play satisfies the Muller condition iff the set of propositions (or the negation of propositions) occurring infinitely often in the play belongs to \mathcal{F} . Given $\mathcal{B} \subseteq \text{AP}_N$, let $\mathcal{F} \upharpoonright \mathcal{B}$ denote the set $\{D \in \mathcal{F} \mid D \subseteq \mathcal{B}\}$. The *Zielonka tree* $\mathcal{Z}_{\mathcal{F}, \mathcal{B}}$ of a Muller condition \mathcal{F} over AP with $\mathcal{B} = \text{AP}_N$ is defined inductively as follows:

1. If $\mathcal{B} \in \mathcal{F}$, then the root of $\mathcal{Z}_{\mathcal{F}, \mathcal{B}}$ is labelled with \mathcal{B} . Let $\mathcal{B}_1, \dots, \mathcal{B}_k$ be all the maximal sets in: $\{\mathcal{B}^* \notin \mathcal{F} \mid \mathcal{B}^* \subseteq \mathcal{B}, \text{ and } \mathcal{B}^* \text{ consistent with respect to } \text{AP}\}$. The root of $\mathcal{Z}_{\mathcal{F}, \mathcal{B}}$ then has as children the Zielonka trees $\mathcal{Z}_{\mathcal{F} \upharpoonright \mathcal{B}_i, \mathcal{B}_i}$ of $\mathcal{F} \upharpoonright \mathcal{B}_i$ for $1 \leq i \leq k$.
2. If $\mathcal{B} \notin \mathcal{F}$, then $\mathcal{Z}_{\mathcal{F}, \mathcal{B}} = \mathcal{Z}_{\overline{\mathcal{F}}, \mathcal{B}}$, where $\overline{\mathcal{F}} = \{D \in 2^{\mathcal{B}} \mid D \notin \mathcal{F} \text{ and } D \text{ is consistent with respect to } \text{AP}\}$.

A node of the Zielonka tree $\mathcal{Z}_{\mathcal{F}, \mathcal{B}}$ is a *Good* node if it is labelled with a set from \mathcal{F} , otherwise it is a *Bad* node.

Equivalent definition of Zielonka trees. We now present an equivalent definition (which suffices for our purposes) of the Zielonka tree $\mathcal{Z}_{\mathcal{F}, \text{AP}_N}$ of a Muller condition \mathcal{F} over AP . Every node of the Zielonka tree $\mathcal{Z}_{\mathcal{F}, \text{AP}_N}$ with is labelled with a consistent subset $\mathcal{B} \subseteq \text{AP}_N$. A node of the Zielonka tree $\mathcal{Z}_{\mathcal{F}, \text{AP}_N}$ is a *Good* node if it is labelled with a set from \mathcal{F} , otherwise it is a *Bad* node. The root is labelled with AP_N . The children of a node v are defined inductively as follows:

1. Suppose v is a Good node labelled with \mathcal{B}_v . Let $\mathcal{B}_1, \dots, \mathcal{B}_k$ be all the maximal sets in: $\{\mathcal{B}^* \notin \mathcal{F} \mid \mathcal{B}^* \subseteq \mathcal{B}_v, \text{ and } \mathcal{B}^* \text{ consistent with respect to } \text{AP}\}$. The node v then has k children (that are all Bad) labelled with $\mathcal{B}_1, \dots, \mathcal{B}_k$.
2. Suppose v is a Bad node labelled with \mathcal{B}_v . Let $\mathcal{B}_1, \dots, \mathcal{B}_k$ be all the maximal sets in: $\{\mathcal{B}^* \in \mathcal{F} \mid \mathcal{B}^* \subseteq \mathcal{B}_v, \text{ and } \mathcal{B}^* \text{ consistent with respect to } \text{AP}\}$. The node v then has k children (that are all Good) labelled with $\mathcal{B}_1, \dots, \mathcal{B}_k$.

The number $m_{\mathcal{F}}$ of a Muller condition. Let \mathcal{F} be a Muller condition that is a consistent subset of 2^{AP_N} . Consider the Zielonka tree $\mathcal{Z}_{\mathcal{F}, \text{AP}_N}$ of \mathcal{F} . We define a number $m_{\mathcal{F}}^v$ for each node v of $\mathcal{Z}_{\mathcal{F}, \text{AP}_N}$ inductively.

$$m_{\mathcal{F}}^v = \begin{cases} 1 & \text{if } v \text{ is a leaf,} \\ \sum_{i=1}^k m_{\mathcal{F}}^{v_i} & \text{if } v \text{ is a Good node and has children } v_1, \dots, v_k, \\ \max\{m_{\mathcal{F}}^{v_1}, \dots, m_{\mathcal{F}}^{v_k}\} & \text{if } v \text{ is a Bad node and has children } v_1, \dots, v_k. \end{cases}$$

The number $m_{\mathcal{F}}$ of the Muller condition \mathcal{F} is defined to be $m_{\mathcal{F}}^{v_r}$ where v_r is the root of the Zielonka tree $\mathcal{Z}_{\mathcal{F}, \text{AP}_N}$.

Lemma 14 ([DJW97]). *Let \mathcal{G}^f be a finite state turn based game. If player 1 has a sure winning strategy for a Muller objective \mathcal{F} from a state s in \mathcal{G}^f , then it has a pure sure winning strategy from s with at most $m_{\mathcal{F}}$ memory states.*

Now we use Zielonka tree analysis to deduce memory requirements of receptive strategies.

- Lemma 15.** 1. Let $\phi_1 = (\diamond \Box F_1) \vee (\diamond \Box F_2) \vee \bigwedge_{i \leq n} (\Box \Diamond I_i)$, where F_1, F_2, I_j are boolean predicates on states of a finite state game \mathcal{G}^f . Player 1 has a pure sure winning strategy from $\text{Sure}_1(\phi_1)$ that requires at most n memory states for the objective ϕ_1 .
2. Let $\phi_2 = (\diamond \Box F) \vee \bigvee_{\alpha \leq m} \left(\diamond \Box F_{\alpha} \wedge \left(\bigwedge_{i \leq n} \Box \Diamond I_{\alpha, i} \right) \wedge \Box \Diamond I_{\alpha} \right)$, where $F, F_{\alpha}, I_{\alpha, i}, I_{\alpha}$ are boolean predicates on states of a finite state game \mathcal{G}^f . Player 1 has a pure sure winning strategy from $\text{Sure}_1(\phi_2)$ that requires at most $(n+1)^m$ memory states for the objective ϕ_2 .

Proof. We present Zielonka tree analysis for each case (in the figures $U = \text{AP}_N$), and use Lemma 14 to deduce the memory bounds. The leaves are depicted with double boundaries in the Figures. Bad nodes are pictured as boxes, and Good nodes as ovals.

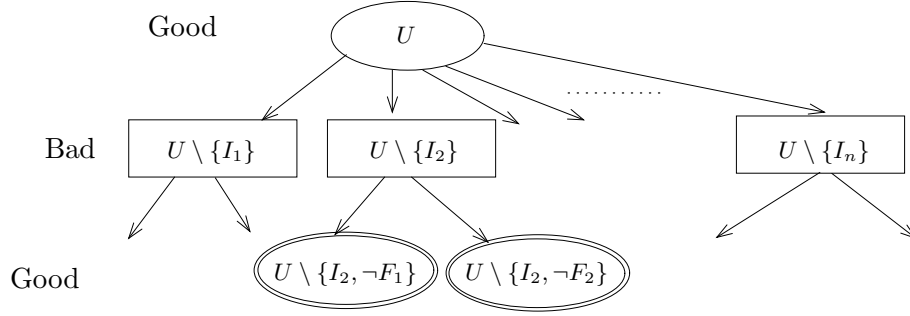


Fig. 5. Zielonka tree for $\phi_1 = (\Diamond \Box F_1) \vee (\Diamond \Box F_2) \vee \bigwedge_{i \leq n} (\Box \Diamond I_i)$.

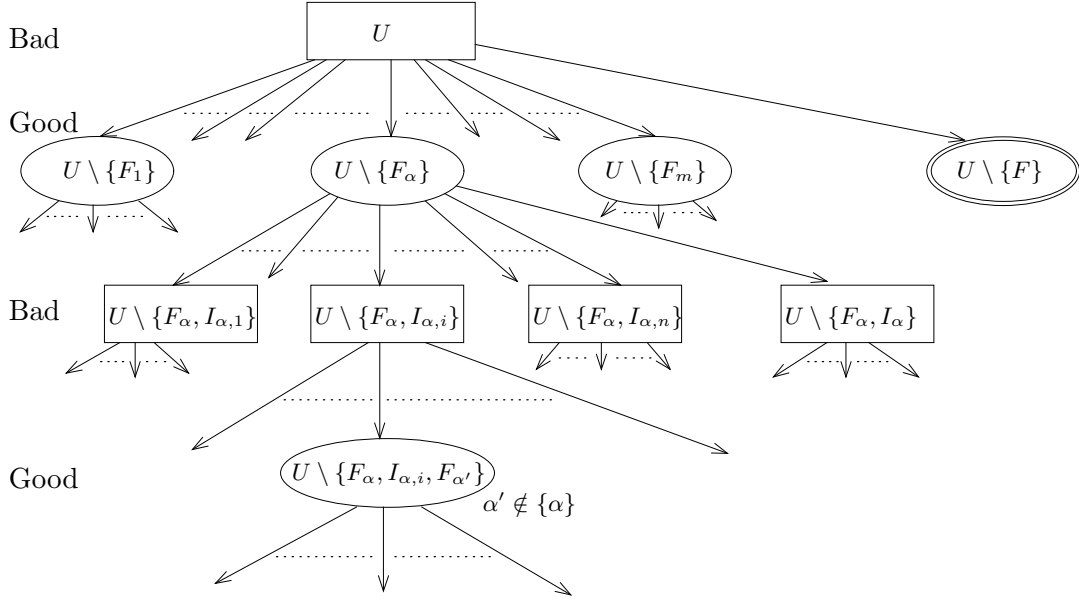


Fig. 6. Zielonka tree for $\phi_2 = (\Diamond \Box F) \vee \bigvee_{\alpha \leq m} (\Diamond \Box F_\alpha \wedge (\bigwedge_{i \leq n} \Box \Diamond I_{\alpha,i}) \wedge \Box \Diamond I_\alpha)$

1. Consider the Zielonka tree in Figure 5. The number m_F^v for the leaf nodes is 1, and also for all the Bad nodes. The number is hence n for root.
2. Consider the (partial) Zielonka tree in Figure 6. The leaves (not shown) are Bad nodes. To compute the m_F^v number for the root, pick an outgoing edge from each Bad node, and retain all edges from Good nodes. For such an edge choice \mathcal{E} , let $\text{Leaf}(\mathcal{Z}_{F, \text{AP}_N}, \mathcal{E})$ denote the number of leaves reachable from the root in the resulting graph. The m_F^v number for the root is then $\max_{\mathcal{E}} (\text{Leaf}(\mathcal{Z}_{F, \text{AP}_N}, \mathcal{E}))$. For the Zielonka tree in Figure 6, let \mathcal{E} be any such edge choice. It can be seen that each Good node in the resulting graph leads to $n + 1$ reachable Good nodes in the next Good level below it. Also, there are m Good levels. Thus the number of leaves reachable from the root in the resulting graph for any \mathcal{E} is $(n + 1)^m$. \square

Corollary 2. Let \mathcal{T} be a timed automaton game with the clocks C , and let $\tilde{\mathcal{T}}$ be the corresponding enlarged game.

1. Let Φ^\dagger be as in Lemma 13. Player 1 has a pure sure winning strategy in $\tilde{\mathcal{T}}$ from $\text{Sure}_1(\Phi^\dagger)$ that requires at most $(|C| + 1)$ memory states.
2. Let Φ^* be as in Lemma 11. Player 1 has a pure sure winning strategy in $\tilde{\mathcal{T}}$ from $\text{Sure}_1(\Phi^*)$ that requires at most $(|C| + 1)^{2^{|C|}}$ memory states.

Proof. For both cases, we first Lemma 15 to the finite game structure $\tilde{\mathcal{T}}^F$ to obtain a pure sure winning strategy $\pi_1^{\tilde{\mathcal{T}}^F}$ in the finite game structure $\tilde{\mathcal{T}}^F$; and then we obtain a pure sure winning strategy $\pi_1^{\tilde{\mathcal{T}}}$ in the game structure

$\ddot{\mathcal{T}}$ by letting $\pi_1^{\ddot{\mathcal{T}}}(\ddot{r}[0..k]) = \langle \Delta, a_1 \rangle$ such that (a) $\pi_1^{\ddot{\mathcal{T}}^F}(\text{Reg}(\ddot{r}[0..k])) = \langle \ddot{R}, a_1 \rangle$, and (b) $\text{Reg}(\ddot{r}[k] + \Delta) = \ddot{R}$. Thus, the memory requirement for a player-1 winning strategy in $\ddot{\mathcal{T}}$ is at as most as that for in the finite game $\ddot{\mathcal{T}}^F$. \square

3.4 Finite Memory Receptive Strategies for Safety Objectives

Player 1 can ensure it stays in a set Y in a receptive fashion if it uses a receptive strategy that only plays moves to Y states at each step. The next theorem uses this fact to characterize safety strategies.

Theorem 2 (Memory requirement for safety). *Let \mathcal{T} be a timed automaton game and $\ddot{\mathcal{T}}$ be the corresponding enlarged game. Let Y be a union of regions of \mathcal{T} . Then the following assertions hold.*

1. $\text{Sure}_1^{\ddot{\mathcal{T}}}(\Box Y) = \underline{\text{Sure}}_1^{\ddot{\mathcal{T}}}((\Box Y) \wedge \Phi^\sharp)$, where $\Phi^\sharp = \Phi^*$ (as defined in Lemma 11), or $\Phi^\sharp = \Phi^\dagger$ (as defined in Lemma 13).
2. Player 1 has a pure, finite-memory, receptive, region strategy in $\ddot{\mathcal{T}}$ that is sure winning for the safety objective $\text{Safe}(Y)$ at every state in $\text{Sure}_1^{\ddot{\mathcal{T}}}(\Box Y)$, that requires at most $(|C| + 1)$ memory states (where $|C|$ is the number of clocks in \mathcal{T}).
3. Player 1 has a pure, finite-memory, receptive, strategy in \mathcal{T} that is sure winning for the safety objective $\text{Safe}(Y)$ at every state in $\text{Sure}_1^{\ddot{\mathcal{T}}}(\Box Y)$, that requires at most $(|C| + 1) \cdot 2^{3 \cdot |C| + 1}$ memory states, i.e. $(\lg(|C| + 1) + 3 \cdot |C| + 1)$ bits of memory (where $|C|$ is the number of clocks in \mathcal{T}).

Proof. 1. (\Leftarrow). If a state $\ddot{s} \in \underline{\text{Sure}}_1^{\ddot{\mathcal{T}}}(\Box Y \wedge \Phi^\sharp)$, then there exists a player-1 winning strategy π_1 such that given any player-2 strategy π_2 , we have that every run \ddot{r} in $\text{Outcomes}(\ddot{s}, \pi_1, \pi_2)$ satisfies both $\Box Y$ and Φ^\sharp . Since Φ^\sharp is satisfies, the strategy π_1 is a receptive strategy by Lemmas 11 and 13. Moreover this strategy ensures that the game stays in Y .

(\Rightarrow). If $\ddot{s} \notin \underline{\text{Sure}}_1^{\ddot{\mathcal{T}}}(\Box Y \wedge \Phi^\sharp)$, then for every player-1 strategy π_1 , there exists a player-2 strategy π_2 such that one of the resulting runs either violates $\Box Y$, or Φ^\sharp . If Φ^\sharp is violated, then π_1 is not a receptive strategy. If $\Box Y$ is violated, then player 2 can switch over to a receptive strategy as soon as the game gets outside Y . Thus, in both cases $\ddot{s} \notin \text{Sure}_1^{\ddot{\mathcal{T}}}(\Box Y)$.

2. The result follows from (a) the first part of the lemma, (b) observing that $\Box Y \wedge \Phi^\sharp$ is an ω -regular objective, (c) Lemma 5, and (d) the first part of Corollary 2 (the memory requirement to ensure $\Box Y \wedge \Phi^\sharp$ is the same as that to ensure Φ^\sharp). We note that the characterization of Lemma 11 for receptive strategies gives a memory bound of $(|C| + 1)^{2^{|C|}}$ for safe receptive strategies.
3. It suffices to show that in the structure \mathcal{T} , player 1 needs only $(3 \cdot |C| + 1)$ bits to maintain the predicates used in the definition of $\ddot{\mathcal{T}}$ in memory. Then, with the help of these $(3 \cdot |C| + 1)$ bits, player 1 can play as if it is playing in $\ddot{\mathcal{T}}$. We assume that player 1 can observe the “flow” during a transition. That is, if the game moves from s to s' in a single game transition, player 1 can observe the “intermediate” states (arising from time passage) “in between” s and s' . Then, player 1 needs only one bit for each of the predicates added to \mathcal{T} in the construcion of $\ddot{\mathcal{T}}$. These bits are updated during the flow of the transition. There are $(3 \cdot |C| + 1)$ predicates.

\square

3.5 Memory Requirement of Receptive Region Strategies for Safety Objectives

We now show memoryless *region* strategies for safety objectives do not suffice (where the regions are as classically defined for timed automata).

Example 3 (Memory necessity of winning region strategies for safety). Consider the timed automaton game \mathcal{T}_3 in Figure 7. The edges a_1^j are player-1 edges and a_2^j player-2 edges. The safety objective of player-1 is to avoid the location “Bad”. It is clear that to avoid the bad location, player-1 must ensure that the game keeps cycling around the locations l_0, l_1, l_2 , and that the clock value of y never exceeds 1. Cycling around only in l_0, l_1 cannot be ensured by a receptive player-1 strategy as player 2 can take smaller and smaller time steps to take the a_2^0

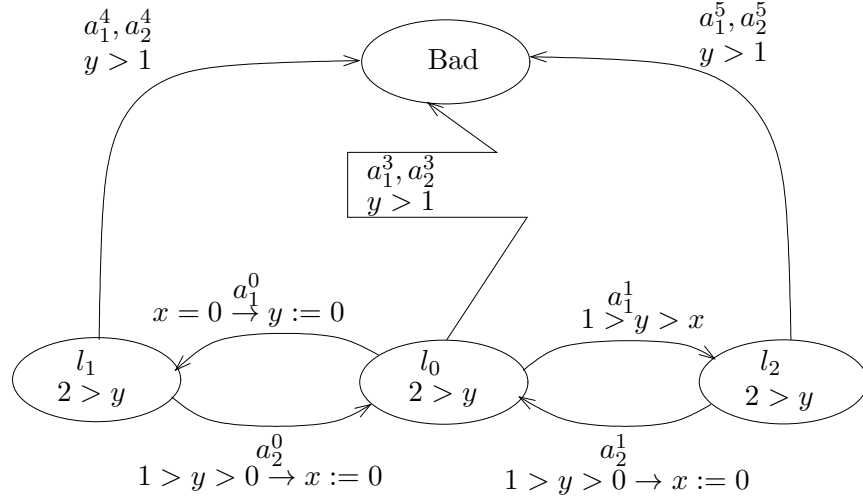


Fig. 7. A time automaton game \mathcal{T}_3 where player-1 does not have receptive region strategies for the safety objective.

transition. Cycling around only in l_0, l_2 also cannot be ensured by a receptive player-1 strategy as the clock value of would always need to stay below 1 without being reset, implying that more than 1 time unit does not pass. Thus, any receptive player-1 strategy which avoids the bad location must cycle infinitely often between l_0, l_1 , and also between l_0, l_2 .

Suppose a player-1 *memoryless* region strategy π_1^* exists for avoiding the bad location, starting from a state in the region $R = \langle l_0, x = 0 \wedge 0 < y < 1 \rangle$. Suppose π_1^* always proposes the transition a_1^0 from the region R_1 . Then, player 2 can take the a_2^0 transitions with smaller and smaller time delays and ensure that the region is R after each a_2^0 transition. This will make time converge, and player 1 will not be blameless, thus π_1^* is not a receptive strategy. Suppose π_1^* always proposes the transition a_1^1 from the region R_1 (or proposes a non-zero time delay move, which has the equivalent effect of disabling the a_1^0 transition). In this case, player 2 can take the a_2^1 transition to again ensure that the region is R after the a_2^1 transition. This will result in the situation where the l_0, l_2 cycle is always taken, time is not divergent, and player 1 is not blameless; thus π_1^* is again not a receptive strategy.

We now demonstrate that a finite-memory (actually memoryless in this case) receptive player-1 strategy π_1^\dagger exists from states in the region $R = \langle l_0, x = 0 \wedge 0 < y < 1 \rangle$ for avoiding the bad location. If the current state is in the region R with the clock value of y being less than $1/2$, then player 1 proposes the a_1^1 transition with a delay which will make make clock y have a value greater than $1/2$. If the current state is in the region R with the clock value of y being greater than or equal to $1/2$, then player 1 proposes to take the a_1^0 transition (immediately). This strategy ensures that against any player-2 receptive strategy: (1) the game will cycle infinitely often between l_0, l_1 , and also between l_0, l_2 , and (2) the clock y will be at least $1/2$ infinitely often, and also be reset infinitely often, giving us time divergence. Thus, π_1^\dagger is a receptive memoryless player-1 winning strategy.

Finally, we demonstrate a player-1 finite-memory receptive *region* strategy π_1^\ddagger for avoiding the bad location, starting from a state in the region $R = \langle l_0, x = 0 \wedge 0 < y < 1 \rangle$. The strategy acts as follows when at region R . If the previous cycle was to l_1 , the strategy π_1^\ddagger proposes to take the edge a_1^1 with a delay which will make make clock y have a value greater than $1/2$. If the previous cycle was to l_2 , the strategy π_1^\ddagger proposes to take the edge a_1^0 (immediately). It can be verified that the strategy π_1^\ddagger requires only one memory state, and is a player-1 winning receptive region strategy. \square

Theorem 3 (Memory necessity of winning region strategies for safety). *There is a timed automaton game \mathcal{T} , a union of regions Y of \mathcal{T} , and a state s such that player 1 does not have a winning memoryless receptive region strategy from s , but has a winning receptive region strategy from s that requires at most $(|C| + 1)$ memory states (where $|C|$ is the number of clocks in \mathcal{T}), for the objective of staying in the set Y .*

Proof. Example 3 presents such a timed automaton game. The memory bound follows from Theorem 2. \square

References

- [AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
- [AH97] R. Alur and T. A. Henzinger. Modularity for timed and hybrid systems. In *CONCUR 97*, Lecture Notes in Computer Science 1243, pages 74–88. Springer, 1997.
- [AM99] E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *HSCC 99*, Lecture Notes in Computer Science 1569, pages 19–30. Springer, 1999.
- [BBL04] P. Bouyer, E. Brinksma, and K. G. Larsen. Staying alive as cheaply as possible. In *HSCC 04*, Lecture Notes in Computer Science 2993, pages 203–218. Springer, 2004.
- [BDMP03] P. Bouyer, D. D’Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In *CAV 03*, Lecture Notes in Computer Science 2725, pages 180–192. Springer, 2003.
- [CDF⁺05] F. Cassez, A. David, E. Fleury, K.G. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR 05*, pages 66–80. Springer, 2005.
- [CHP08] K. Chatterjee, T. A. Henzinger, and V. S. Prabhu. Trading infinite memory for uniform randomness in timed games. In *HSCC 08*, Lecture Notes in Computer Science 4981. Springer, 2008.
- [dAFH⁺03] L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *CONCUR 03*, Lecture Notes in Computer Science 2761, pages 144–158. Springer, 2003.
- [DJW97] S. Dziembowski, M. Jurdziński, and I. Walukiewicz. How much memory is needed to win infinite games? In *LICS 97*, pages 99–110. IEEE Computer Society, 1997.
- [DM02] D. D’Souza and P. Madhusudan. Timed control synthesis for external specifications. In *STACS 02*, Lecture Notes in Computer Science 2285, pages 571–582. Springer, 2002.
- [HK99] T. A. Henzinger and P. W. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221:369–392, 1999.
- [HP06] T. A. Henzinger and V. S. Prabhu. Timed alternating-time temporal logic. In *FORMATS 06*, Lecture Notes in Computer Science 4202, pages 1–17. Springer, 2006.
- [PAMS98] A. Pnueli, E. Asarin, O. Maler, and J. Sifakis. Controller synthesis for timed automata. In *Proc. System Structure and Control*. Elsevier, 1998.
- [SGSAL98] R. Segala, R. Gawlick, J.F. Søggaard-Andersen, and N. A. Lynch. Liveness in timed and untimed systems. *Inf. Comput.*, 141(2):119–171, 1998.
- [Tho97] W. Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.